INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

U·M·I

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Order Number 9137056

**Robust controller design for lightly damped systems with
feedback delay**

Hung, Donald Lu-Cheng, Ph.D.

Case Western Reserve University, 1991

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

ROBUST CONTROLLER DESIGN FOR LIGHTLY
DAMPED SYSTEMS WITH FEEDBACK DELAY

By

DONALD LU-CHENG HUNG

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Thesis Advisor: Dr. Stephen M. Phillips

DEPARTMENT OF ELECTRICAL ENGINEERING AND APPLIED PHYSICS
CASE WESTERN RESERVE UNIVERSITY

JANUARY 1991

CASE WESTERN RESERVE UNIVERSITY
GRADUATE STUDIES

We hereby approve the thesis of

candidate for the _____

degree.*

Signed:

Stephen M. Phillips
(Chairman)

Sheldon G. Rubin
Walter M. Rubin
Kenneth A. Logan

Date _____

*We also certify that written approval has
been obtained for any proprietary material
contained therein.

I grant to Case Western Reserve University the right to use this work, irrespective of any copyright, for the University's own purposes without cost to the University or to its students, agents and employees. I further agree that the University may reproduce and provide single copies of the work, in any format other than in or from microforms, to the public for the cost of reproduction.

Donald Lu-cheng Jung

ROBUST CONTROLLER DESIGN FOR LIGHTLY DAMPED SYSTEMS WITH FEEDBACK DELAY

Abstract

By

DONALD LU-CHENG HUNG

Stability analysis based on an idealized robot-stiff environment model shows that for a lightly damped system, a time delay in the feedback may become a critical issue to the system's stability if the delay is nonnegligible with respect to the system's natural frequency. The attention of this thesis focuses on the design of controllers for lightly damped systems with nonnegligible feedback delay. A generalized control system framework is adopted and two major assumptions are made: the plant and the controller are linear, time-invariant (*LTI*), and the plant is open-loop stable. The time delay in the feedback loop is modeled in the plant. By discretizing, the perturbed plant remains as *LTI* and the sample-hold effects have been taken into consideration. A

closed-loop transfer matrix obtained from the control system framework contains all closed-loop transfer functions of interest. Therefore the closed-loop performance specifications for controller design can be expressed as constraints or objectives for each entry of this transfer matrix. Via a bilinear transformation, all stable closed-loop transfer matrices form a convex set and many performance specifications become convex functions defined on the set. This property allows a convex programming approach for the controller design procedure. Since this has to be done through a numerical approach, the solution space is truncated to form a finite dimensional Euclidean subspace. The designed controllers are robust with respect to time delays in the feedback action. Other robustness considerations, such as unmodeled plant dynamics, can also be taken into the design specifications. Since the convex program is a global optimal technique, once a controller is found, it is optimal in the sense that it meets the design specifications. On the other hand, if no solution is found then there exists no *LTI* controller which can satisfy the design specifications. Simulation results show that the designed controllers can effectively stabilize the lightly damped system and significantly improve its performance.

Acknowledgements

I would like to express my sincere gratitude to my advisor, Professor Stephen Phillips for his thoughtful guidance, persistent patience and tremendous help throughout the development of this thesis.

Sincere thanks are also extended to the members of my thesis committee, Professor Kenneth Loparo and Professor Wyatt Newman, for their enlightening discussion, generous encouragement and constructive criticism. I would like to thank the department chairman, Professor Sheldon Gruber, for serving on the committee and the continuous support from the Electrical Engineering Department.

Special thanks go to my previous advisor, Professor Harry Mergler, for all the advice, help and support he gives me during these years.

Finally, I thank my family for their loving support, understanding and encouragement. I shall always be grateful.

Contents

Abstract	ii
Acknowledgements	iv
List of Tables	viii
List of Figures	xiii
1. Introduction	1
1.1 Background and Motivation	1
1.2 Objective and Outline	7
2. The Effect of Delayed Feedback on Stability of Lightly Damped Systems	10
2.1 The Idealized Force Feedback Control	10
2.2 Negative Force Feedback with Delay	14
2.3 Positive Force Feedback with Delay	17
2.4 Summary	21
3. Control System Configuration and Modeling of Feedback De- lay	23

3.1	Control System Configuration	23
3.1.1	The Framework	23
3.1.2	The Controller Design Problem	25
3.1.3	An Example	27
3.2	Modeling the Feedback Delay	31
3.2.1	The Digital Controller	31
3.2.2	Modeling the Delay in the Plant	34
3.2.3	Discretizing — A Practical Approach	38
4.	Controller Design — A Convex Programming Approach.....	42
4.1	Outline of the Controller Design Method	42
4.2	Performance Specification as Convex Functions	50
4.2.1	Performance Specifications for Steady State Properties	50
4.2.2	Performance Specifications for Transient Response . . .	52
4.2.3	Bounded Quantities in Time Domain	55
4.2.4	Bounds in Frequency Domain	56
4.3	Parameterization of the Closed-loop Transfer Matrix and All Stabilizing Controllers	62
4.4	Finite Dimensional Approximation of the Solution Space . . .	68
5.	A Design Example.....	72
5.1	The System	72

5.2	Stable Coprime Factorizations	75
5.3	Choice of Nominal Controller	77
5.4	Design Tools	80
5.5	Some Results	84
5.6	The Controller	93
6.	Conclusions	103
6.1	Summary	103
6.2	Main Features and Advantages	103
6.3	Main Disadvantages	104
6.4	Further Research	105
A.	Related Mathematical and Control Concepts	106
A.1	The Ring Concept	106
A.2	Matrix Coprimeness and the Bezout Identity	107
A.3	Convexity — Sets, Functions and Programming Problems	109
A.4	Concept of H_∞ and $\ H\ _\infty$	110
A.5	The Small Gain Theorem [Zames 66]	113

List of Tables

5.1	Coefficients of of the reduced-order controller.	101
5.2	Poles and zeros of the reduced-order controller.	102

List of Figures

2.1	The simplified robot-environment model.	11
2.2	Bode plot for system with feedback delay – negative feedback.	20
2.3	Bode plot for system with feedback delay – positive feedback.	21
3.1	The closed-loop control system.	24
3.2	A simple force control system.	27
3.3	The simple force control system with noise inputs.	28
3.4	The simple force control system in generalized framework.	29
3.5	The closed-loop system with a digital controller.	31
3.6	The control sequence generated by idealized digital controller.	32
3.7	The closed-loop system in which the controller delay is modeled in the digital controller.	35
3.8	The delayed control sequence obtained by inserting a delay unit in the controller.	36
3.9	The closed-loop system in which the delay unit has been moved out of the controller.	37
3.10	Control sequences of the system in Fig. 3.9 at the D/A output and plant input.	38

3.11	The closed-loop system in which the controller delay is modeled in the plant.	39
3.12	The closed-loop control system with feedback delay.	40
4.1	An illustration of the controller design method.	48
4.2	Specification at unit step response $r(t)$ by upper and lower en- velopes	53
4.3	A typical closed-loop system.	57
4.4	Illustration of M-circle radius.	58
4.5	Closed-loop system with plant perturbation.	59
4.6	Equivalence of Fig. 4.5 seen by Δ	60
4.7	Converting a multiplicative plant perturbation into an additive perturbation.	61
5.1	The force control system with feedback delay.	73
5.2	Positions of the open-loop poles.	79
5.3	Step response of the open-loop system.	80
5.4	Root-locus of the closed-loop system with delayed negative force feedback ($0 \leq K \leq 1$).	81
5.5	Step-response of the closed-loop system with delayed negative force feedback ($K = 0.01$).	82
5.6	Root-locus of the closed-loop system with delayed positive force feedback ($-1 \leq K \leq 0$).	83

5.7	Step response of the closed-loop system with delayed positive force feedback ($K = -0.1$).	84
5.8	The specification envelopes and the step response corresponding to a barely stabilizing nominal controller K_{nom}	85
5.9	The actuator effort response to a unit step force command with the nominal controller.	86
5.10	The endpoint force response to a unit step force command with the Design 1 controller.	87
5.11	The actuator effort response to a unit step force command with the Design 1 controller.	88
5.12	The frequency response of H_{13} with the Design 1 controller. .	89
5.13	The endpoint force step responses (solid line: Design 1 controller; dotted line: nominal controller).	90
5.14	The actuator effort step responses (solid line: Design 1 controller; dotted line: nominal controller).	91
5.15	The endpoint force response to a unit step force command with the Design 2 controller.	92
5.16	The actuator effort response to a unit step force command with the Design 2 controller.	93
5.17	The frequency response of H_{13} with the Design 2 controller. .	94

5.18	The endpoint force step responses (solid line: Design 2 controller; dotted line: nominal controller).	95
5.19	The actuator effort step responses (solid line: Design 2 controller; dotted line: nominal controller).	95
5.20	The endpoint force response to a unit step force command with the Design 3 controller.	96
5.21	The actuator effort response to a unit step force command with the Design 3 controller.	96
5.22	The frequency response of H_{13} with the Design 3 controller. . .	97
5.23	The endpoint force step responses (solid line: Design 3 controller; dotted line: nominal controller).	97
5.24	The actuator effort step responses (solid line: Design 3 controller; dotted line: nominal controller).	98
5.25	The frequency response of H_{13} with the nominal controller. . .	98
5.26	The magnitude response of $K_1(e^{j\Omega})$ (solid line: original controller; dotted line: reduced-order controller).	99
5.27	The phase response of $K_1(e^{j\Omega})$ (solid line: original controller; dotted line: reduced-order controller).	99
5.28	The magnitude response of $K_2(e^{j\Omega})$ (solid line: original controller; dotted line: reduced-order controller).	100

5.29	The phase response of $K_2(e^{j\Omega})$ (solid line: original controller; dotted line: reduced-order controller).	100
5.30	The poles of the reduced-order controller.	101
5.31	The poles of the closed-loop transfer functions.	102
A.1	Illustration of $\ H\ _\infty$ norm.	112
A.2	A simple system for interpreting $\ H\ _\infty$ norm.	113
A.3	The loop for explaining the small gain theorem.	114

Chapter 1

Introduction

1.1 Background and Motivation

Due to the attempts to apply robots in assembly operations and manufacturing tasks, robot force control becomes an increasingly active research area. For a number of years, considerable effort has been made in the development of this area. Basic force control strategies such as explicit force control[Nevins and Whitney 73], stiffness control[Salisbury 80], damping control[Whitney 77, Mason 81], impedance control[Hogan 85] and hybrid control[Paul and Shimano 76, Raibert and Craig 81] have been proposed in the literature. Further developments based on these concepts are made by many researchers[Shin and Lee 85, West and Asada 85, Yoshikawa 86, Kazerooni and Sheridan and Honpt 86, Kazerooni 87, Aderson and Spong 87, Wang and Kuo 88, Goldenberg 88, Schutter 88]. Adaptive control[Koivo 86, Honehangi and Koivo 87, Slotine and Li 87a-b, Fukuda, Kitamura and Tanie 87, Liu, Chang and Zhang 88], sliding mode control[Slotine and Sastry 83, Slotine 85] and control based on differential geometric theory[Tarn, Bejezy and Yun 88] have also been applied to robot

force control.

Despite of the richness in proposed force control strategies, succesful implementations are rare. A general problem encountered in robot force control is *instability*. A widely observed phenomenon is that when in contact with rigid environments, manipulators violently chatter and lose contact with the environments[Wlassich 86, Hogon 87, An and Hollerbach 87a].

More than a decade ago, Whitney[Whitney 77] analyzed damping control stability based on a simplified discrete model. A tradeoff between the force feedback gain, environment stiffness and sampling rate was found. For many years after Whitney's paper, stability issues related to force control were not taken into serious consideration by many of the researchers. In the last few years, however, a lot more attention has been given to this subject. Kazerooni[Kazerooni 87, Kazerooni and Tsay 88a-b] proposes stability criteria for a generalized impedance control scheme. He claims that in order to guarantee stability, there must be some initial compliance either in the manipulator or in the environment. Colgate and Hogan[Colgate and Hogan 87a-b] propose a necessary and sufficient condition for the stability of a linear robot system (possibly including feedback loops) coupled to a passive environment. They point out that the robot system must present a passive admittance if it is to be stable in contact with arbitrary passive impedances. Yabuta et. al.[Yabuta and Chonu and Beni 88] clarify the conditions for global asymptotic stability

of the hybrid control scheme. They claim that the velocity feedback gain must be positive definite and for a given object stiffness, the position and force feedback gains must be selected such that the Hessian of the potential function is positive definite. Some fundamental and practical issues have also been studied or noticed, e.g., modelling of constrained robot [Hemami and Wyman 79, Orin and Oh 81, McClamroch and Huang 85, McClamroch 86, Huang 86, Huang and McClamroch 87], force control dynamic and kinematic instability [An and Hollerbach 87a-b, Zhang 89], instability due to unmodeled dynamics in robot systems and sensor, actuator bandwidth limitations [Eppinger and Seering 86, 87, 88, 89]. Experimental results in robot force control are reported as well [Wu and Paul 80, Khatib and Burdick 86, Wlassich 86, Youcef-Toumi and Li 87, An, Atkeson and Hollerbach 88, Elosegui, Daniel and Sharkey 90].

The proposed control strategies or criteria seldom successfully remedy the stability problem in robot force control. Some practical approaches and suggestions are:

- Introducing passive compliance [Whitney 87, An and Hollerbach 87a, Kazerooni and Tsay 88b], either on the robot side (e.g., using soft sensor) or on the environment side (e.g., using soft cover).
- Introducing damping based on knowledge of the environmental stiffness [Khatib and Burdick 86, An and Hollerbach 87a, Yabuta, Chonu and Beni 88, Hamilton 88].

- Using joint torque feedback instead of end-point force feedback[Wu and Paul 80, Salisbury 80, An and Hollerbach 87a, Vertal 89, Elosegui and Daniel and Sharkey 90].

For any control system, the paramount requirement is *stability*. Without stable behavior, any performance requirement becomes meaningless. The widespread observation of instability in robot force control strongly suggests that there is a need to identify and solve fundamental problems related to such systems[Whitney 87]. In some recently conducted robot force control experiments, the adoption of a new generation of robots based on direct drive technology brings an opportunity for fundamental studies. The elimination of gears in direct drive manipulators eliminates, or significantly alleviates the concomitant problems such as backlash, friction and joint flexibility. The consequences are improved performance (speed and accuracy) and controllability, and more importantly, the significant reduction of unmodeled dynamics in the robot system. These improvements lead to the exposure of some fundamental, but not well recognized problems that can have a profound effect on the behavior of force controlled robots.

An and Hollerbach's experiments[An and Hollerbach 87a] on the third link of the MIT Serial Link Direct Drive Arm (DDArm) is especially valuable in providing physical insight to force control. An important result from a previous study[Eppinger and Seering 86, 87, 88, 89] was that sensor-actuator

non-collocation (i.e., the existence of unmodeled dynamics between the sensor and the actuator due to the flexibility in the robot system) could lead to instability. An and Hollerbach[An and Hollerbach 87a] and Wlassich[Wlassich 86] consider non-collocation to be the main cause of the instability appearing in their experiments. However, for the MIT DDArm, a single mass rigid body is a very accurate model for its third link (although it is impossible to completely eliminate the unmodeled dynamics). The unstable behavior of this link under force control suggests that the unmodeled dynamics may not be the only source which causes instability.

The search for other possible causes of instability for force control is also stimulated by the lack of agreement among existing hypotheses. One example is the successful use of low pass filters in both forward and force feedback loops in An and Hollerbach's experiments[An and Hollerbach 87a] contradicting Eppinger and Seering's conclusion[Eppinger and Seering 87]. Another example is that Wlassich's important observation in implementing impedance control[Wlassich 86], that when the robot encounters a stiff environment, instability manifests itself when the desired mass is less than the actual robot mass. This phenomenon cannot be explained adequately by the unmodeled dynamics.

Due to the increasing computation power of control processors, the existence of a time delay in the feedback loop is rarely considered (Among voluminous publications on robot force control, very few [Lawrence 88, Aderson and Spong 89] address this problem). In most analyses of a computer controlled system, the control action is assumed to be simultaneous with the sensing. Further, the analysis is frequently based on a continuous system ignoring the effects of sampling and data reconstruction (hold circuit). A common misunderstanding is that today's control processors are fast enough so that the controller delay is negligible [Eppinger and Seering 87]. The minimization of unmodeled dynamics including friction and rotor inertia in the MIT DDArm, together with the knowledge of the stiffness of rigid environments obtained by An and Hollerbach [An and Hollerbach 87a] based on the DDArm's force controlled experiments, reveal that the coupled robot-stiff environment form a highly underdamped system. Our study shows that for such lightly damped systems, when the feedback loop is closed, time delay is a critical factor in determining the system's stability. If the system's natural frequency is large, a very small feedback delay can cause a significant phase shift which may significantly deteriorate the stability of the close-loop controlled system. Based on this understanding, the development of controllers which are robust to the feedback delay becomes the motivation of this thesis.

1.2 Objective and Outline

The objective of this thesis is to study controller design methods for closed-loop control systems with *inherent, nonnegligible* time delays in the feedback control actions.

To help understand the effect of feedback delay on stability of lightly damped systems, we start with a case study — a simplified model of a force controlled manipulator. Despite the simplicity of the model, it is close to the theoretical model of the third link of the MIT DDArm used in reported experiments [An and Hollerbach 1987a]. Some results obtained from their experiments are therefore used in our discussion. We find that even for today’s fast processors, the time delay in force feedback is a nontrivial issue for the stability of force controlled robots. To focus our attention on the effect of feedback delay, the existence of sensor-actuator non-collocation is not considered. Further, it is assumed that the manipulator is initially rest on the environment. In this case, if the coupled system is stable under control, the manipulator should not lose contact with the environment. The “lose contact” phenomenon due to non-zero initial velocity or instability of the coupled system, is not within the scope of this study.

In the study of controller design, no specific configuration for the plant is assumed, but it does assume that the plant can be modeled as a linear, time invariant (*LTI*), open-loop stable system (scalar or multiple input-multiple

output). This assumption is a strong restriction and its adoption is based on the following considerations:

- The well developed theory in *LTI* systems and *LTI* controller design lend us tools and provide guide lines for our study.
- Linear control system often form the core or basis of control systems designed for nonlinear systems, e.g., feedback linearized, gain scheduled or adaptive controllers.
- There has been serious effort in eliminating nonlinearities(whether the nonlinearities belong to the plant or not) in control systems, e.g., the use of compensation, feedback linearization, direct drive technology, better design, etc. .
- Some control applications are regulator problems, i.e., the goal is to keep the plant output near some desired values. The plant may be well modeled as a *LTI* system in the vicinity of these desired points.

The *LTI* plant model is discretized and the effect of sample-hold and time delay in feedback control are considered in the discretized model. The *LTI* controller is obtained via an optimization approach. The unmodeled dynamics, as well as some other robustness issues, have also been taken into consideration.

The remainder of this thesis is organized as follows:

Chapter 2 discusses the existence of time delays in the feedback loop and its effect on stability of lightly damped systems. Chapter 3 describes the control system structure and the modeling of the feedback time delay in the structure. Chapter 4 presents the controller design method in detail. Chapter 5 applies the results developed in Chapter 3 and Chapter 4 to a force control problem. Chapter 6 concludes the thesis and discusses the advantages and disadvantages of the controller design method as well as possible extension of this work. Related mathematics background and control concepts are included in the Appendix.

Chapter 2

The Effect of Delayed Feedback on Stability of Lightly Damped Systems

In this chapter, we discuss the existence of time delays in feedback control, as well as their effect on the stability of lightly damped systems. The discussion is based on the stability study of a simplified robot-environment model under force control although the conclusions are applicable to general.

2.1 The Idealized Force Feedback Control

The simplified robot-environment model is shown in Figure 2.1, where the manipulator is modeled as a lumped mass m_r , the environment is modeled as a linear spring with stiffness k_e , and a linear viscous damper with a damping coefficient b_e . The contact force is commonly modeled as $k_e x$. The system therefore has the dynamic equation:

$$m_r \ddot{x} + b_e \dot{x} + k_e x = f_a \quad (2.1)$$

where the actuator force f_a is determined by specific control laws.

Figure 2.1: The simplified robot-environment model.

The environment is assumed to be stiff. To ease the discussion, we assign the following parameter values:

$$m_r = 1 \text{ kg}, b_e = 0.1 \text{ kg/sec}, k_e = 6 \times 10^4 \text{ N/m}.$$

The knowledge of k_e is based on An et.al.'s experiment results[An and Atkeson and Hollerbach 88]. They find that the stiffness of aluminum is well above $6 \times 10^4 \text{ N/m}$, but geometric information of the material under test is not provided.

Explicit Force Control. The simplest implementation of explicit force control is to use a proportional controller with the control law:

$$f_a = k_f f_d - k_f k_e x \quad (2.2)$$

Note that with the conventional feedback ($k_f > 0$), the control effort based on the measured contact force $k_e x$ always point to the norm outward direction of the environment. We refer to this as *negative* force feedback. Substitute the above control law into Eq. 2.1 to obtain the closed-loop dynamic equation:

$$m_r \ddot{x} + b_e \dot{x} + (1 + k_f)k_e x = k_f f_d \quad (2.3)$$

Eq. 2.3 shows that the effect of conventional *negative* force feedback is to increase the stiffness of the environment. With the knowledge that the value of stiffness k_e of rigid environment is very large while the damping b_e is very small, it is clear that the manipulator against the stiff environment under idealized explicit force control is a highly underdamped system with high natural frequency. The result is high frequency oscillations of the system.

Impedance Control. In impedance control, the closed-loop dynamic equation is specified by a desired mechanic impedance with desired mass m_d , damping k_v and stiffness k_p :

$$m_d \ddot{x} + k_v(\dot{x} - \dot{x}_d) + k_p(x - x_d) = -f_i \quad (2.4)$$

where, $\dot{x}_d$ and x_d are the desired velocity and virtual position, respectively; f_i is the contact force and can be modeled as $k_e x$ as usual. For simplicity, here we let $b_e = 0$, the control law then can be derived as

$$f_a = \frac{m_r}{m_d} [k_v(\dot{x}_d - \dot{x}) + k_p(x_d - x)] - \left(\frac{m_r}{m_d} - 1\right) k_e x \quad (2.5)$$

Note that the objective of impedance control is to make $m_d < m_r$. In this case, according to our previous definition, the control effort based on the contact force $k_e x$ is a *negative* feedback. On the other hand, if we let $m_d > m_r$, then the control effort based on $k_e x$ always point to the norm inward direction of the environment and we refer to this as *positive* force feedback. Substituting the control law into Eq. 2.1 yields the closed-loop dynamic equation:

$$m_d \ddot{x} + k_v \dot{x} + (k_p + k_e)x = k_v \dot{x}_d + k_p x_d \quad (2.6)$$

Compare the above equation with the desired impedance Eq. 2.4, notice that in the idealized case $f_i = k_e x$, it turns out that idealized impedance control makes the manipulator under control possesses exactly the same dynamic behavior as the desired mechanical impedance, no matter whether *negative* feedback ($m_r > m_d$) or *positive* feedback ($m_r < m_d$) is applied. The closed-loop dynamic equation Eq. 2.6 also shows that if the environment stiffness is high, the coupled system is highly underdamped unless the desired damping coefficient k_v is adapted based on the knowledge of k_e .

Note that for the idealized case here, the measurement of the contact force is assumed to be perfect and the force feedback is assumed to be instantaneous. Under these assumptions, force feedback increases the stiffness of the coupled system but does not introduce instability. In the real world however, from the moment of the contact to the moment of generating a control action based on the measured contact force, there are always time delays due to the A/D

and D/A conversion time, sensor, actuator and power amplifier bandwidth limitations, as well as the software computational time. Among these, the A/D conversion time and the computational time produce pure delays. In modern implementations, the absolute values of the time delay can be made very small. However, the amount of time delay the system can tolerate is also dependent on the characteristics of the system itself. The following sections will discuss more about the tolerable delays.

2.2 Negative Force Feedback with Delay

Last section shows that, the simplified robot-environment model under idealized force control is always stable; but if the environment is stiff, the coupled system is highly underdamped. In this section, we consider delayed *negative* force feedback, the sensed contact force is assumed to be accurate, but the control action is delayed by τ .

Explicit Force Control. Let $k_e x(t - \tau)$ substitute $k_e x$ in control law Eq. 2.2 and then use Eq. 2.1, the closed-loop dynamic equation with delayed feedback action is obtained:

$$m_r \ddot{x}(t) + b_e \dot{x}(t) + k_e x(t) + k_f k_e x(t - \tau) = k_f f_d \quad (2.7)$$

The characteristic equation of the above is

$$s^2 + \frac{b_e}{m_r} s + \frac{k_e}{m_r} + \frac{k_f k_e}{m_r} e^{-\tau s} = 0 \quad (2.8)$$

For small delay τ , using the first-order rational approximation $e^{-\tau s} \approx \frac{1}{1+\tau s}$, stability conditions can be found through the Routh-Hurwitz Criterion:

$$\tau > 0, 1 + \frac{b_e}{m_r}\tau > 0, \frac{b_e}{m_r} + \frac{k_e}{m_r}\tau > 0, (1 + k_f)\frac{k_e}{m_r} > 0 \quad (2.9)$$

$$\frac{k_f k_e}{m_r}\tau - \left(\frac{b_e}{m_r}\right)^2 \tau + \frac{b_e k_e}{m_r^2}\tau^2 < \frac{b_e}{m_r} \quad (2.10)$$

For *negative* force feedback, $k_f > 0$, stability conditions in Formulas 2.9 are always satisfied. With the given parameter values of the coupled system, and knowing that the delay τ can be less than 5×10^{-3} sec., it is resonable to neglect the second and third terms on the left hand side of Inequality 2.10, so the tolerable delay for stability is approximately:

$$\tau < \frac{b_e}{k_f k_e} \text{ sec.} \quad (2.11)$$

Impedance Control. Let $k_e x(t - \tau)$ substitute $k_e x$ in control law Eq. 2.5 and use Eq. 2.1, the closed-loop dynamic equation with feedback delay is:

$$m_d \ddot{x}(t) + k_v \dot{x}(t) + (k_p + \frac{m_d}{m_r} k_e) x(t) + (1 - \frac{m_d}{m_r}) k_e x(t - \tau) = k_p x_d + k_v \dot{x}_d \quad (2.12)$$

The above 2.12 has the characteristic equation

$$s^2 + \frac{k_v}{m_d} s + \left(\frac{k_p}{m_d} + \frac{k_e}{m_r}\right) + \frac{k_e}{m_d} \left(1 - \frac{m_d}{m_r}\right) e^{-\tau s} = 0 \quad (2.13)$$

Again, using the first-order rational approximation for small delay τ and applying the Routh-Hurwitz Criterion to find the stability conditions:

$$\tau > 0, 1 + \frac{k_v}{m_d}\tau > 0, \frac{k_v}{m_d} + \left(\frac{k_p}{m_d} + \frac{k_e}{m_r}\right)\tau > 0, \frac{k_p + k_e}{m_d} > 0 \quad (2.14)$$

and

$$k_e \left(\frac{1}{m_d} - \frac{1}{m_r} \right) \tau - \left(\frac{k_v}{m_d} \right)^2 \tau - \frac{k_v}{m_d} \left(\frac{k_p}{m_d} + \frac{k_e}{m_r} \right) \tau^2 < \frac{k_v}{m_d} \quad (2.15)$$

Stability conditions in Formulas 2.14 are always satisfied. In Inequality 2.15, knowing the ranges of values for k_e (of stiff environments) and τ , noticing that the original goal of impedance control (*negative* force feedback) is to make the robot nimble by setting the desired mass m_d less than the robot mass m_r , also providing that k_v is not adapted based on the knowledge of k_e , the second and third terms of the left and side of Inequality 2.15 can be neglected, and the tolerable delay for stability is closely approximated by:

$$\tau < \frac{m_r}{m_r - m_d} \frac{k_v}{k_e} \text{ sec.} \quad (2.16)$$

Both conditions 2.11 and 2.16 indicate that the larger the environment stiffness, the smaller feedback delay that can be tolerated. This is consistent with the experience that soft sensors or soft covers on environments improve stability because both efforts reduced the effective stiffness k_e [An, Atkeson and Hollerbach 1988]. Condition 2.11 also shows that the tolerance of feedback delay for explicit force control will be very limited if damping in the system is small. For impedance control, Condition 2.16 shows that unless the desired mass m_d is very close to the actual robot mass m_r , or the desired damping effect k_v is adapted based on the knowledge of k_e , the tolerance of feedback delay is also very limited. As examples, with the given parameter values, for explicit force control, if the feedback gain $k_f = 0.01$ (negative feedback), then

from Eq. 2.11 the tolerate delay is only 1.67×10^{-4} seconds; for impedance control, if $\frac{m_d}{m_r} = 0.5$ (negative feedback), $k_v = 7.05 \text{ N} \cdot \text{s/m}$ and $k_p = 50 \text{ N/m}$, the tolerate delay is 2.35×10^{-4} seconds.

2.3 Positive Force Feedback with Delay

In the previous section, the characteristic equations Eq. 2.8 and Eq. 2.13 can be put in the generalized form:

$$s^2 + as + b + ce^{-\tau s} = 0 \quad (2.17)$$

where coefficient c relates to feedback force. For explicit force control, $c = \frac{k_f k_e}{m_r}$; for impedance control, $c = \frac{k_e}{m_d} (1 - \frac{m_d}{m_r})$. It is clear that when *positive* force feedback is applied ($k_f < 0$ in explicit force control or $\frac{m_d}{m_r} > 1$ in impedance control), the coefficient c changes sign which indicates the change of direction of the feedback action. A careful look at the stability conditions Formulas 2.10 and Formulas 2.15 shows that under *positive* force feedback control, these two conditions are always satisfied. However, the tolerable delay time cannot be obtained via the same approach as we did in the previous section, since the rational approximation for the delay breaks down. Therefore in finding tolerable delay for *positive* force feedback, a graphical stability margin approach is adopted. The generalized characteristic equation (2.17) can be rearranged as

$$1 + \frac{N(s)}{D(s)} e^{-\tau s} = 0 \quad (2.18)$$

where, $N(s) = c$, $D(s) = s^2 + as + b$. The stability properties of the coupled system can be studied by virtue of the Bode plot of $\frac{N(s)}{D(s)}e^{-\tau s}$. If, with no delay,

$$\frac{N(\omega)}{D(\omega)} = M(\omega)\angle\phi(\omega)$$

where $M(\omega)$ and $\phi(\omega)$ are gain and phase of $\frac{N(\omega)}{D(\omega)}$, respectively. Then for the system with delay,

$$\frac{N(\omega)}{D(\omega)}e^{-\tau s} = M(\omega)\angle\phi(\omega) - \omega\tau\frac{180^\circ}{\pi}$$

Note that the gain is not changed while the delay $e^{-\tau s}$ produces an extra phase angle which is continuously decreasing with frequency, therefore the phase plot may have multiple phase crossovers which can be determined by

$$\phi(\omega) - \omega\tau\frac{180^\circ}{\pi} = \pm(2n+1)180^\circ, \text{ for } n = 0, 1, 2, \dots \quad (2.19)$$

where, $\pm(2n+1)180^\circ$ represents the negative real axis in the correspondent Nyquist plot. In order to check stability by looking at the gain margin at phase crossovers, rewrite Eq. 2.19 as

$$\phi(\omega) = \omega\tau\frac{180^\circ}{\pi} \pm(2n+1)180^\circ, \text{ for } n = 0, 1, 2, \dots \quad (2.20)$$

When linear scale of ω is used, Eq. 2.20 means that with delay $e^{-\tau s}$, the new phase crossovers are the intersections of the original phase plot $\phi(\omega)$ and a group of straight lines. See Figure 2.2, which is for the case of conventional

negative force feedback. For *positive* force feedback, the coefficient c in generalized characteristic equation 2.17 changes its sign so that Eq. 2.18 becomes

$$1 - \frac{N(s)}{D(s)}e^{-\tau s} = 0 \quad (2.21)$$

Note the fact that, if $H(\omega) = |H(\omega)|\angle\psi(\omega)$, then $-H(\omega) = |H(\omega)|\angle\psi(\omega) \pm 180^\circ$, i.e., with *positive* force feedback, the gain is remain unchanged and the phase is shifted for $\pm 180^\circ$ (the sign here can be optionally chosen, it changes the starting point of the phase plot but does not affect the analysis). The phase crossovers are now determined by

$$\phi(\omega) \pm 180^\circ = \omega\tau \frac{180^\circ}{\pi} \pm (2n+1)180^\circ, \text{ for } n = 0, 1, 2, \dots \quad (2.22)$$

This process is illustrated by Figure 2.3. In which the phase shift due to *positive* force feedback is considered $+180^\circ$. Since the phase here is continuously decreasing with frequency, to find the range of tolerable delay, we need to look at the two concecutive intersections with straight lines $\omega\tau \frac{180^\circ}{\pi} + 180^\circ$ and $\omega\tau \frac{180^\circ}{\pi} - 180^\circ$. Clearly, if the phase shift due to *positive* feedback is considered as -180° , then the two intersections we need to look at are with straight lines $\omega\tau \frac{180^\circ}{\pi} - 180^\circ$ and $\omega\tau \frac{180^\circ}{\pi} - 540^\circ$. In Fig. 2.3, the first intersection on phase plot at $(0, 180^\circ)$ does show that, when using *positive* force feedback, $|c| < b$ in generalized characteristic equation 2.17 is required for stability. For explicit force control, this requirement is equivalent to $k_f > -1$, which is also noted from stability condition 2.9; for impedance control, $|c| < b$ is always satisfied

if $m_d > m_r$.

Figure 2.2: Bode plot for system with feedback delay - negative feedback.

With the help of above two figures, we are now able to find the tolerable delay τ_2 for *positive* force feedback based on knowledge of tolerable delay τ_1 for *negative* force feedback. From Fig. 2.2 and Fig. 2.3, we have

$$\phi(\omega) = \frac{\omega_c \tau_1}{\pi} 180^\circ - 180^\circ \quad (2.23)$$

and

$$\phi(\omega) + 180^\circ = \frac{\omega_c \tau_2}{\pi} 180^\circ - 180^\circ \quad (2.24)$$

for *negative* and *positive* force feedback, respectively. Where ω_c is the frequency at gain crossover. Solving τ_2 from Eq. 2.23 and 2.24, we have

$$\tau_2 = \tau_1 + \frac{\pi}{\omega_c} \quad (2.25)$$

Figure 2.3: Bode plot for system with feedback delay – positive feedback.

With the parameter values given by the example, ω_c can be found as 316 rad/sec. Using Eq. 2.25, the tolerable delay τ_2 is found as about $\tau_1 + 10$ ms. Therefore *positive* force feedback provides a significant extension to the tolerable delay for stability.

2.4 Summary

We list the main points discussed in this chapter. Note that although the control-related conclusions are drawn from a force controlled manipulator, they apply to all lightly damped systems.

- When the robot encounters a stiff environment, the coupled system is highly underdamped and the system's natural frequency is very large.
- The feedback action based on the sensed signal is always delayed from the moment the signal occurred.
- Due to the highly underdamped and high natural frequency of the system under control, a very small (absolute value) delay in the feedback action may cause a large phase lag (relatively) which significantly deteriorates the stability of the system.
- The underdamped system is much more tolerable to the delay in feedback action under *positive* feedback control than under *negative* feedback control.

Chapter 3

Control System Configuration and Modeling of Feedback Delay

We concluded in the previous chapter that a time delay in the feedback action is a crucial issue for stability of lightly damped systems. In this chapter, we introduce a control system configuration which is adopted from [Boyd et al 88] and forms the basis of our study. We describe the configuration first, then show how to take the feedback delay into account by modeling it inside the *plant*.

3.1 Control System Configuration

3.1.1 The Framework

The generalized framework for closed-loop control systems is shown in Figure 3.1.

In the above framework, P is the *plant*, K is the *controller*, the negative sign follows the tradition of *negative* feedback, i.e., if K is a proportional controller and has a *positive* value, then the feedback is *negative*, w , u , z , y

Figure 3.1: The closed-loop control system.

are *input/output vectors*, they are explicitly defined as follows:

- The *exogenous input vector* w consists of command signals and disturbance (noise) signals.
- The *control input vector* u consists of signals that are generated by the controller.
- The *regulated output vector* z consists of signals which are tunable or of interest in the closed-loop system.
- The *sensed output vector* y consists of signals that are accessible to the controller.

Notice that according to the definition, the components of different input/output vectors can be nested. For instance, if a command is accessible by the controller (e.g., a set point), then it should appear in both vector w and vector y ; on the other hand, since control signals in u are tunable, they could also be included in vector z .

Since the definitions for the input/output vectors are detailed and exact, with the generalized framework shown in Figure 3.1, a specific control problem can be made explicit: how and where the command and noise signals act upon the plant; which signals are regulated; which signals are generated by the controller, etc. . The consequence of the exactness is: one can specify control objectives explicitly based on the overall consideration of performance and internal stability requirements. This will be made clear in later discussions.

3.1.2 The Controller Design Problem

In Chapter 1 we have indicated that in this study, we assume the plant P is linear, time invariant (LTI) and open-loop stable. As a sequel of this assumption, the plant is able to be described by a transfer matrix. Each entry of the matrix is a transfer function from one of the inputs (the elements of w or u) to one of the outputs (the elements of z or y). The plant transfer matrix P can be partitioned as

$$P = \begin{bmatrix} P_{zw} & P_{zu} \\ P_{yw} & P_{yu} \end{bmatrix} \quad (3.1)$$

where P_{zw} is the transfer matrix from w to z , P_{zu} is the transfer matrix from u to z , P_{yw} is the transfer matrix from w to y , and P_{yu} is the transfer matrix from u to y . The open-loop system therefore can be written as

$$\begin{bmatrix} z \\ y \end{bmatrix} = \begin{bmatrix} P_{zw} & P_{zu} \\ P_{yw} & P_{yu} \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix}$$

We assume further that controller K is also *LTI* so that it can be described by a transfer matrix as well. Based on the control framework shown in Figure 3.1, the closed-loop transfer matrix from w to z can be derived as

$$H_{zw} = P_{zw} - P_{zu}K(I + P_{yu}K)^{-1}P_{yw} \quad (3.2)$$

The entries of the transfer matrix H_{zw} are the closed-loop transfer functions from each exogenous input to each regulated variable, for example, the closed-loop transfer function from some disturbance to some controlled variable, from some sensor to some regulated internal variable, from some command to some actuator, etc. It is now clear that if we make vectors w and z include every signal about which we intend to express a constraint or specification, then H_{zw} should contain every closed-loop transfer function of interest to us. The closed-loop transfer functions, as indicated by Eq. 3.2, depend on the controller K , and the controller design problem then can be described as:

Design a controller K , which stabilizes the transfer matrix H_{zw} (which means all of its entries are stable transfer functions), and make the desired closed-loop transfer functions (entries of H_{zw}) satisfy certain performance specifications.

The topic of controller design will be further discussed in detail in Chapter 5.

3.1.3 An Example

To illustrate how the control system configuration works, here we adopt a simple force control structure from [An and Hollerbach 87a], as shown in Figure 3.2.

Figure 3.2: A simple force control system.

In Fig. 3.2, f_d is the command force, f_i is the contact force, K is the controller, f_a is the actuator output (for simplicity, the actuator here is assumed to have a unit gain), R is the dynamic model of the coupled robot-environment system. The force command feedforward loop is used to eliminate the steady-state error. Assuming there exist sensor and actuator noises, we can redraw the control system diagram as below:

Figure 3.3: The simple force control system with noise inputs.

In Fig. 3.3, n_s is the sensor noise, n_r is the actuator noise, f_s is the sensed force, u is the control signal. The system in Fig. 3.3 can be easily arranged to fit in the generalized framework (Fig. 3.1). To see the flexibility of this framework, suppose the actuator effort is constrained, so we place f_a in vector z , the result is shown in Fig. 3.4.

Figure 3.4: The simple force control system in generalized framework.

According to Fig. 3.4, the exogenous input vector w consists of force command f_d , actuator noise n_τ and sensor noise n_s :

$$w = \begin{bmatrix} f_d \\ n_\tau \\ n_s \end{bmatrix}$$

the regulated output vector z consists of the contact force f_i and the actuator effort f_a :

$$z = \begin{bmatrix} f_i \\ f_a \end{bmatrix}$$

The control input vector u is degenerated into a scalar since the system has a single actuator, and the sensed output vector y consists of signals accessible

to the controller — the force command f_d and the sensed force f_s :

$$y = \begin{bmatrix} f_d \\ f_s \end{bmatrix}$$

The plant, which has four inputs and four outputs, is shown by the dashed block in Fig. 3.4. The transfer matrix of the plant is:

$$P = \begin{bmatrix} P_{zw} & P_{zu} \\ P_{yw} & P_{yu} \end{bmatrix} = \begin{bmatrix} R & R & 0 & R \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ R & R & 1 & R \end{bmatrix} \quad (3.3)$$

The transfer matrix of controller K is a one by two matrix:

$$K = \begin{bmatrix} K_1 & K_2 \end{bmatrix}$$

then from Eq. 3.2, the closed-loop transfer matrix H_{zw} is

$$H_{zw} = \begin{bmatrix} \frac{R(1-K_1)}{1+RK_2} & \frac{R}{1+RK_2} & \frac{-RK_2}{1+RK_2} \\ \frac{1-K_1}{1+RK_2} & \frac{1}{1+RK_2} & \frac{-K_2}{1+RK_2} \end{bmatrix} \quad (3.4)$$

The second row of H_{zw} consists of the closed-loop transfer functions from the force command and noises to the actuator output, and therefore relate to the constraints on the actuator effort. We may limit the “size” of these transfer functions in some appropriate sense in our design specifications which will be discussed in later chapters.

So far we have described the control system configuration, explained its framework and posed the controller design problem. The next section discusses the problem of how to take the feedback delay into consideration by modeling it in this framework.

3.2 Modeling the Feedback Delay

3.2.1 The Digital Controller

Because of the fact that robots are computer controlled, we resketch the generalized framework for closed-loop control systems which was described by Fig. 3.1, as below:

Figure 3.5: The closed-loop system with a digital controller.

In the above Fig. 3.5, the controller K (the dashed box) is implemented by a computer. The D/A converter can be alternatively included in the plant P , since like power amplifiers and actuators, the effect of D/A converters is similar to a continuous lag component. Note that in discrete-time system analysis, the commonly used *stroboscopic model* only considers the zero-order-hold effect, the pure delay due to the computation and A/D conversion time is not considered, i.e., it assumes that the control signal u based on the plant output y measured at time t_k will act at exactly the same time, as illustrated by Fig. 3.6.

Figure 3.6: The control sequence generated by idealized digital controller.

However, in implementing the controller, the real sequence of operation in the computer can be described by the following two cases:

Case 1. Sequence of operation:

1. Read inputs from A/D.
2. Compute control variable.
3. Execute other tasks.
4. Write output to D/A
5. Go back to step 1.

Case 2. Sequence of operation:

1. Read inputs from A/D.
2. Compute control variable.
3. Write output to D/A.
4. Execute other tasks.
5. Go back to step 1.

For both Case 1 and Case 2, the procedure “execute other tasks” can mean some routine management, test, computation for other loops, etc. . For dedicated controllers, it can be the part of the control computation (for the next control output) that can be done after the current control signals been calculated and before the next sensor read takes place. The difference between the two cases is: In Case 1, the computer finishes all the computation and routine tasks first, then outputs control signals right before the next cycle; in Case

2, the computer outputs control signals as soon as they have been calculated. Obviously, for Case 1, there is a delay τ which is equal to the sampling period T , while for Case 2 delay τ is less than T . The third case, $\tau > T$, can also occur if we purposely postpone the output of calculated control signals. For all cases, a pure delay due to the computation and A/D conversion time required by the digital controller, must be taken into consideration. In our later discussions, we name this pure delay as “controller delay”.

3.2.2 Modeling the Delay in the Plant

In order to take the controller delay into account, we can modify the system shown in Fig. 3.5 by inserting a pure delay unit τ in the controller, as shown in Fig. 3.7.

For the system shown in Fig. 3.7, the sequence of control signal u with respect to measured plant output y is depicted by Fig. 3.8, where the computational delay is assumed to be shorter than the sampling period.

Note that the control sequence is viewed at the D/A converter output, or equivalently, at the plant input. We now rearrange Fig. 3.7 by placing the delay unit τ at the output of the D/A converter, as shown in Fig. 3.9.

Compare Fig. 3.9 with Fig. 3.7, a substantial change has been made — the delay unit has been moved out of the controller, therefore the controller again becomes an idealized one. Fig. 3.10 indicates that for the system shown

Figure 3.7: The closed-loop system in which the controller delay is modeled in the digital controller.

in Fig. 3.9, the controller generates ideal control sequences, while the plant accepts delayed control sequences (u' in Fig. 3.9). In other words, from the plant's point of view, the two systems described by Fig. 3.7 Fig. 3.9 are equivalent.

Based on Fig. 3.9 we can merge the plant P and the delay unit τ , the result is shown in Fig. 3.11, where $\tilde{P}$ is the *perturbed* plant which takes the delay issue into account.

We are now ready to go back to the original control framework (Fig. 3.1). Based on the above discussion, the closed-loop control system with time delay in the feedback loop can be sketched in Fig. 3.12.

Figure 3.8: The delayed control sequence obtained by inserting a delay unit in the controller.

In Fig. 3.12, the dashed block represents the perturbed plant $\tilde{P}$. Since it consists of a pure delay unit τ and the *LTI* plant P , $\tilde{P}$ can still be described by a transfer matrix:

$$\tilde{P} = \begin{bmatrix} P_{rw} & \tilde{P}_{ru} \\ P_{yw} & \tilde{P}_{yu} \end{bmatrix}$$

P_{zu} and P_{yu} , two of the submatrices of the original plant transfer matrix (Eq. 3.1), have been perturbed due to the feedback delay.

For the example given in Subsection 3.1.3, with the consideration of feedback delay, the original plant transfer matrix Eq. 3.3 is perturbed and becomes

Figure 3.9: The closed-loop system in which the delay unit has been moved out of the controller.

$$\tilde{P} = \begin{bmatrix} P_{zw} & \tilde{P}_{zu} \\ P_{yw} & \tilde{P}_{yu} \end{bmatrix} = \begin{bmatrix} R & R & 0 & \tilde{R} \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ R & R & 1 & \tilde{R} \end{bmatrix} \quad (3.5)$$

and the closed-loop transfer matrix Eq. 3.4 becomes

$$H_{zw} = \begin{bmatrix} \frac{R - \tilde{R}K_1}{1 + \tilde{R}K_2} & \frac{R}{1 + \tilde{R}K_2} & \frac{-\tilde{R}K_2}{1 + \tilde{R}K_2} \\ \frac{1 - K_1 - (R - \tilde{R})K_2}{1 + \tilde{R}K_2} & \frac{1}{1 + \tilde{R}K_2} & \frac{-K_2}{1 + \tilde{R}K_2} \end{bmatrix} \quad (3.6)$$

In Eq. 3.6, $\tilde{R}$ is the perturbed open-loop transfer function of the robot-environment system.

Figure 3.10: Control sequences of the system in Fig. 3.9 at the D/A output and plant input.

The feedback delay has now been modeled in our framework, the remaining problem — how to mathematically handle the delay, will be explained in the next subsection.

3.2.3 Discretizing — A Practical Approach

The theory of continuous-time systems with time delays is complicated because the systems are infinite dimensional. However, the delay can be handled easily for discrete-time systems. There are some tools ready to use:

If a continuous-time *LTI* plant is given in the following state-space form

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (3.7)$$

Figure 3.11: The closed-loop system in which the controller delay is modeled in the plant.

$$y(t) = Cx(t) + Du(t)$$

under digital control, with the consideration of the effects of zero-order-hold and a time delay τ (less than the sampling period T) in the control action, the discrete *LTI* model of the plant described by Eq. 3.7 is:

$$\begin{bmatrix} x(kT + T) \\ u(kT) \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma_1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(kT) \\ u(kT - T) \end{bmatrix} + \begin{bmatrix} \Gamma_0 \\ I \end{bmatrix} u(kT) \quad (3.8)$$

where,

$$\Phi = e^{AT} \quad (3.9)$$

Figure 3.12: The closed-loop control system with feedback delay.

$$\Gamma_0 = \int_0^{T-\tau} e^{A^T s} \cdot B \quad (3.10)$$

$$\Gamma_1 = e^{A(T-\tau)} \int_0^{\tau} e^{A^T s} \cdot B \quad (3.11)$$

For the case where the time delay τ is greater than T , let

$$\tau = (d-1)T + \tau', \quad 0 < \tau' < T, \quad d \text{ integer},$$

the discrete-time *LTI* model of the plant is :

$$\begin{bmatrix} x(kT + T) \\ u(kT - dT + T) \\ \vdots \\ u(kT - T) \\ u(kT) \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma_1 & \Gamma_0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & I \\ 0 & 0 & 0 & \cdots & I \end{bmatrix} \begin{bmatrix} x(kT) \\ u(kT - dT) \\ \vdots \\ u(kT - 2T) \\ u(kT - T) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ I \end{bmatrix} u(kT) \quad (3.12)$$

where, Γ_0 and Γ_1 are given by Eq. 3.10 and Eq. 3.11 with τ replaced by τ' .

See [Åström and Wittenmark 84].

In Z domain, the equivalent tool is the *delayed Z-transform*, it is also known as the *modified Z-transform*. See [Barker 52] and [Jury 56].

Except the consideration of avoiding mathematical difficulties, the discrete-time approach is a natural choice since we are dealing with a problem which uses digital control. A discrete model is of course more appropriate in describing characteristics related to digital control, such as the effect of zero-order-hold. The biggest advantage of the discrete-time approach, is that the resulting discrete-time transfer functions, i.e., the entries of the discrete-time plant transfer matrix $P(z)$ and the closed-loop transfer matrix $H_{zw}(z)$, are maintained as rational functions (of z). This is the key issue that allows us to use the mature theory developed for linear systems to tackle our problem of systems with inherent time delays.

Chapter 4

Controller Design — A Convex Programming Approach

In the previous chapters, we pointed out that the importance of feedback delay to stability of lightly damped systems; we discussed how to model the feedback delay and insert it into the plant model; we arranged to maintain our system with delay LTI by discretizing it; we also showed that our control configuration has made it possible to include *every* closed-loop transfer function of interest to be available as some element of the closed-loop transfer matrix H_{zw} so that all design specifications can be expressed explicitly as requirements on the closed-loop transfer matrix H_{zw} . Based on these results, this chapter will detail the controller design method.

4.1 Outline of the Controller Design Method

We have seen in Chapter 3 that our control configuration allows us to include every signal of interest in the exogenous input vector w or in the regulated output vector z . Therefore, the $n_{exog} \times n_{reg}$ closed-loop transfer matrix

$H_{zw} = P_{zw} - P_{zu}K(I + P_{yu}K)^{-1}P_{yw}$ contains every interesting closed-loop transfer function between two certain points in the system. We use $\mathcal{H}$ to denote the set of all closed-loop transfer matrices achievable by varying K , the controller.

As indicated before, the most important issue in the controller design is to stabilize the closed-loop transfer matrix H_{zw} , i.e., to guarantee each entry of H_{zw} to be a stable closed-loop transfer function. We denote all achievable stable closed-loop transfer matrices as $\mathcal{H}_{stable}$, a subset of $\mathcal{H}$:

$$\mathcal{H}_{stable} = \{H_{zw} \in \mathcal{H} \mid H_{zw} = P_{zw} - P_{zu}K(I + P_{yu}K)^{-1}P_{yw} \text{ for all stabilizing } K\}$$

and $\mathcal{H}_{stable} \subseteq \mathcal{H}$.

Considering stability as a premise, any performance specification which is related to some specific entry of the closed-loop transfer matrix H_{zw} , specifies a subset of $\mathcal{H}_{stable}$ which contains all achievable H_{zw} that are stable and meet the given performance specification. For instance, if H_{ij} , the i, j th entry of H_{zw} is a closed-loop transfer function whose input is some noise, we want to make sure it is stable and also limit its “size” by not allowing its $\|H\|_{\infty}$ norm to exceed a given value c . Corresponding to this specification, the set $\{\mathcal{H}_{stable} \mid \|H_{ij}\|_{\infty} < c\}$ contains all the qualified closed-loop transfer matrices. We denote the i th performance specification as ps_i and its associated subset of $\mathcal{H}_{stable}$ as $\mathcal{H}_i$, thus for a design problem which has up to l performance specifications, the set of stable closed-loop transfer matrices that satisfy all l

performance specifications is

$$\mathcal{H}_{desired} = \mathcal{H}_1 \cap \dots \cap \mathcal{H}_l.$$

The task of controller design then can be considered as seeking an appropriate controller K so that an $H_{zw} \in \mathcal{H}_{desired}$ can be achieved. Clearly, a single infeasible performance specification or any two mutually inconsistent performance specifications will cause $\mathcal{H}_{desired}$ empty. For simplicity, when there is no ambiguity we will omit the subscript of H_{zw} and denote $\mathcal{H}_{stable}$ as

$$\mathcal{H}_{stable} = \{H \mid H \text{ stable}\}. \quad (4.1)$$

Here we point out two important observations that form the basis of this thesis:

1. If H_{zw} can be put in a *free parameter representation* form in which it is *affine* with respect to a single *free variable*, then $\mathcal{H}_{stable}$ is a *convex set*, i.e. (refer to definitions given in A.3), for any two closed-loop transfer matrices $H^{(1)} \in \mathcal{H}_{stable}$ and $H^{(2)} \in \mathcal{H}_{stable}$,

$$H^{(3)} = \alpha H^{(1)} + (1 - \alpha) H^{(2)} \quad (0 \leq \alpha \leq 1)$$

is also in $\mathcal{H}_{stable}$.

2. If $\mathcal{H}_{stable}$ is a *convex set*, then many performance specifications can be described by *convex functions* on $\mathcal{H}_{stable}$, that is,

$$ps_i = \Phi_i(H), \quad (4.2)$$

for $i = 1, 2, \dots$, number of convex performance specifications, and when both $H^{(1)}$ and $H^{(2)}$ are in $\mathcal{H}_{stable}$, then

$$\Phi_i(\alpha H^{(1)} + (1 - \alpha)H^{(2)}) \leq \alpha \Phi_i(H^{(1)}) + (1 - \alpha)\Phi_i(H^{(2)})$$

The two observations above will be made clear later on in this chapter. The importance of these observations is in that the *convexity* property of both $\mathcal{H}_{stable}$ and ps suggests the possibility to arrange the controller design problem into a *convex programming problem*. Compared to many local optimization techniques, the convex programming is a global optimization technique, the advantage of a global optimization approach is that whenever a solution is found, it is a global solution (may not be unique); on the other hand, if no solution is found, it simply means there is *no* solution for the given performance specifications, i.e., the set $\mathcal{H}_{desired}$ is empty.

According to the above discussion, in order to make the *convex programming* controller design approach feasible, the first step we have taken, is the *free parameter representation* of the closed-loop transfer matrix H_{zw} , as shown below:

$$H_{zw} = T_1 + T_2 Q T_3 \tag{4.3}$$

where T_1 , T_2 and T_3 are “fixed” matrices that only depend on the *LTI* plant and the *nominal controller* K_{nom} — an arbitrary controller which stabilizes P_{yu} but usually does not satisfy the given performance specifications (notice

that $\mathcal{H}_{stable}$ is always nonempty, i.e., K_{nom} always exist), while Q is the “free parameter” matrix which is related to the controller K via a bilinear transformation. In Section 4.3 we will show that by using the bilinear mapping between Q and K , all closed-loop transfer matrices can be represented in the form of Eq. 4.3, further, as long as Q is stable, H_{zw} is stable. Thus the set of achievable stable closed-loop transfer matrices (Eq. 4.1) can be expressed as

$$\mathcal{H}_{stable} = \{T_1 + T_2 Q T_3 \mid Q \in \mathbf{S}^{n_{ctrl} \times n_{sens}}\} \quad (4.4)$$

where $\mathbf{S}^{n_{ctrl} \times n_{sens}}$ is the set of all $n_{control-signal} \times n_{sensed-signal}$ matrices with stable transfer functions as its elements.

With the free parameterization of H_{zw} , we now verify the convexity of $\mathcal{H}_{stable}$. If $H^{(1)} = T_1 + T_2 Q^{(1)} T_3$ and $H^{(2)} = T_1 + T_2 Q^{(2)} T_3$ are in $\mathcal{H}_{stable}$, then

$$\begin{aligned} H^{(3)} &= \alpha H^{(1)} + (1 - \alpha) H^{(2)} \\ &= \alpha(T_1 + T_2 Q^{(1)} T_3) + (1 - \alpha)(T_1 + T_2 Q^{(2)} T_3) \\ &= T_1 + T_2[\alpha Q^{(1)} + (1 - \alpha) Q^{(2)}] T_3 \\ &= T_1 + T_2 Q^{(3)} T_3 \end{aligned} \quad (4.5)$$

From Eq. 4.5, two facts are evident: firstly, $H^{(3)}$ is an *achievable* closed-loop transfer matrix by varying the controller K , since $Q^{(3)}$ is achievable and a bilinear mapping between Q and K exists; secondly, when both $Q^{(1)}$ and $Q^{(2)}$ are in $\mathbf{S}^{n_{ctrl} \times n_{sens}}$, $Q^{(3)} = \alpha Q^{(1)} + (1 - \alpha) Q^{(2)}$ is also in $\mathbf{S}^{n_{ctrl} \times n_{sens}}$ (recall that $\mathbf{S}^{m \times n}$ is a *ring*), in other word, the set

$$\mathcal{Q} = \{Q \in \mathbf{S}^{n_{ctrl} \times n_{sens}}\} \quad (4.6)$$

is *convex*, so $H^{(3)}$ is also *stable* and therefore $H^{(3)} \in \mathcal{H}_{stable}$.

As a consequence, the convexity of performance specifications are also secured and ps_i in Eq. 4.2 becomes a *convex functional* of Q , i.e.,

$$ps_i = \Phi_i(H) = \tilde{\Phi}_i(Q), \quad Q \in \mathbf{S}^{n_{ctrl} \times n_{sens}} \quad (4.7)$$

for $i = 1, 2, \dots$, number of convex performance specifications and when both $Q^{(1)}$ and $Q^{(2)}$ are in $\mathcal{Q}$, then

$$\tilde{\Phi}_i(\alpha Q^{(1)} + (1 - \alpha)Q^{(2)}) \leq \alpha \tilde{\Phi}_i(Q^{(1)}) + (1 - \alpha)\tilde{\Phi}_i(Q^{(2)})$$

The controller design problem now becomes: adjust Q to achieve an $H_{zw} \in \mathcal{H}_{desired}$, then go back to determine the corresponding controller K via the bilinear transformation. This idea is illustrated by Fig. 4.1.

Compare Eq. 4.3 with Eq. 3.2, the original representation of H_{zw} . Note that in Eq. 3.2 representation, H_{zw} depends on the controller K in a *linear fractional* fashion such that a relatively simple performance specification (i.e., a simple constraint on H_{zw}) will correspond to a very complicated constraint on the controller K and it is less obvious that the linear combination of two closed-loop transfer matrices is also a closed-loop transfer matrix which can be obtained by simply adjusting the controller K . Thus, without the free parameter representation of H_{zw} , the convex programming approach is not straightforward.

Fig. 4.1 shows that the controller design idea resembles a dual port network:

Figure 4.1: An illustration of the controller design method.

the parameters inside the network are fixed, but by changing the impedance at one port, the equivalent impedance seen from the other port will change. In our case, the nominal controller K_{nom} and the plant parameters are fixed, as Q varies over the entire set $\mathcal{Q} = \{Q \in \mathbb{S}^{n_{ctrl} \times n_{sens}}\}$, the equivalent controller seen from the control input u and the measured output y sweeps out all possible stabilizing controllers which correspond to all achievable closed-loop stable transfer matrices that form the set $\mathcal{H}_{stable}$.

The reader may notice that the convex sets $\mathcal{H}_{stable}$ and $\mathcal{Q}$ are described by *infinite dimensional vector spaces*, i.e., stable matrices H_{zw} or Q are considered as points in the spaces. Since there is no known method to analytically solve the infinite dimensional convex programming problem related to our design, a

second step we have taken is to convert our problem into a *finite dimensional Euclidean Space* to enable the numerical approach. This is done by approximating entries of the matrices T_1, T_2, T_3 and Q as stable *FIR* filters with large but finite number of delays. Coefficients of the *FIR* filters associated with Q are the *decision variables* to be determined. In Section 4.4 we will discuss this approach in more detail and show that the sets

$$\bar{\mathcal{Q}} = \{\bar{Q} \mid \text{finite dimensional approximation of } Q\}$$

and

$$\mathcal{X} = \{\mathbf{x} \mid \text{vectors of coefficients of FIR filters in associate with } \bar{Q}\} \quad (4.8)$$

are *convex*. Therefore the performance specifications can be presented as functionals of $\mathbf{x}$:

$$ps_i = \Phi_i(H) = \tilde{\Phi}_i(Q) = \hat{\Phi}_i(\mathbf{x}), \quad (4.9)$$

for $i = 1, 2, \dots$. In the above equations, $\mathcal{X}$ forms a finite dimensional Euclidean Space. When the optimum $\mathbf{x}^*$ is found, $\bar{Q}$ is formed and the correspondent controller K can be obtained through the bilinear transformation.

Knowing that the free parameterization of H_{zw} exists (the proof can be found later in Section 4.3), Eq. 4.5 verifies the convexity of the set $\mathcal{H}_{stable}$. In the rest of this chapter, Section 4.2 discusses the representation of performance specifications as convex functions; Section 4.3 discusses the free parameterization of all closed-loop transfer matrices via the bilinear mapping between Q

and K ; the finite dimensional approximation of the solution space is discussed in Section 4.4. The remaining question: how to actually set up and implement the *convex programming problem*, will be explained in Chapter 5 through an example.

4.2 Performance Specification as Convex Functions

In this section we show that many performance specifications can be described as *convex functions* of the closed-loop transfer matrix H_{zw} (again for simplicity we will omit the subscript when there is no ambiguity). We neither intend to say that all performance specifications are convex functions of H , nor try to list all performance specifications that are describable by convex functions of H . What we want to show is that a broad class of performance specifications can be described as convex functions of H .

4.2.1 Performance Specifications for Steady State Properties

Many performance specifications for steady state properties can be handled in the frequency domain. For example, the i, j th entry of H_{zw} is a closed-loop transfer function from the j th input to the i th output. If the input is some

command and the output is the regulated variable supposed to be controlled by the command, the performance specification of *asymptotic tracking* of unit step inputs means the step response of H_{ij} must converge to one. By applying the *final value theorem*, the specification is equivalent to

$$H_{ij}(z)|_{z=1} = 1 \quad (4.10)$$

On the contrary, if we wish the i th regulated variable to asymptotically reject a constant command from the j th input channel, or if the j th input is a disturbance, then the performance specification become *asymptotic decoupling* or *asymptotic regulation*. For both cases, the frequency domain representation is

$$H_{ij}(z)|_{z=1} = 0 \quad (4.11)$$

The performance specification in Eq. 4.10 and Eq. 4.11 can be generalized as

$$\Phi(H) = H_{ij}(z)|_{z=1} = c \quad (4.12)$$

where $H \in \mathcal{H}_{stable}$ and c is a constant.

We will see $\Phi(H)$ satisfies the definition of a *convex function* described by Eq. A.6. Since for $H^{(1)}$ and $H^{(2)}$ in $\mathcal{H}_{stable}$,

$$\begin{aligned} \Phi(\lambda H^{(1)} + (1 - \lambda)H^{(2)}) &= [\lambda H_{ij}^{(1)}(z) + (1 - \lambda)H_{ij}^{(2)}(z)]|_{z=1} \\ &= \lambda H_{ij}^{(1)}(z)|_{z=1} + (1 - \lambda)H_{ij}^{(2)}(z)|_{z=1} \\ &= \lambda \Phi(H^{(1)}) + (1 - \lambda)\Phi(H^{(2)}) \\ &= \lambda c + (1 - \lambda)c = c \end{aligned} \quad (4.13)$$

In the above discussion, if the input is a more complicated signal, we may still be able to handle it by applying techniques of limit analysis in addition to the *final value theorem*. For instance, *asymptotic tracking* of ramp inputs can be specified by two functions:

$$\begin{aligned}\Phi_1(H) &= H_{ij}(z)|_{z=1} = 1, \\ \Phi_2(H) &= \frac{d}{dz}H_{ij}(z)|_{z=1} = 0, \quad H \in \mathcal{H}_{stable}.\end{aligned}$$

It can be easily verified that both Φ_1 and Φ_2 are *convex functions* of H .

4.2.2 Performance Specifications for Transient Response

Performance specifications for transient response, such as overshoot, undershoot, rise time and settling time for step response, are best specified in the time domain. An alternative (also in the time domain) is to specify the envelopes of the responses. For example, if $r(t)$ is the unit step response of the closed-loop transfer function H_{ij} , let $r_{max}(t) = 1 + c_1 e^{-a_1 t}$ and $r_{min}(t) = 1 - c_2 e^{-a_2 t}$ as the upper and lower envelopes, respectively, then the required transient response can be specified as

$$r_{min}(t) \leq r(t) \leq r_{max}(t) \text{ for } t \geq 0, \quad (4.14)$$

as shown in Fig. 4.2.

Time domain performance specifications such as Eq. 4.14 can be discretized

Figure 4.2: Specification at unit step response $r(t)$ by upper and lower envelopes

and converted into a finite number of constraints, e.g.,

$$r_{\min}(k) \leq r(k) \leq r_{\max}(k) \text{ for } k = 0, 1, 2, \dots, N. \quad (4.15)$$

Since by the *inversion formula*,

$$r(k) = \frac{1}{2\pi j} \oint_{\Gamma} R(z) z^{k-1} dz.$$

where $R(z) = \frac{z}{z-1} H_{ij}(z)$ for unit step input, the performance specifications in Eq. 4.15 can be described in Z domain:

$$c_1 \leq \frac{1}{2\pi j} \oint_{\Gamma} \frac{H_{ij}(z)}{z-1} z^k dz \leq c_2 \text{ for } k = 0, 1, \dots, N. \quad (4.16)$$

where, constants c_1 and c_2 have the values of the envelopes $r_{\min}$ and $r_{\max}$ at

time k . Now, let

$$\Phi_k(H) = \frac{1}{2\pi j} \oint_{\Gamma} \frac{H_{ij}(z)}{z-1} z^k dz \text{ for } k = 0, 1, \dots, N. \quad (4.17)$$

and we find that they are all *convex functions*, since

$$\begin{aligned} \Phi_k(\lambda H^{(1)} + (1-\lambda)H^{(2)}) &= \frac{1}{2\pi j} \oint_{\Gamma} \frac{\lambda H_{ij}^{(1)}(z) + (1-\lambda)H_{ij}^{(2)}(z)}{z-1} z^k dz \\ &= \lambda \frac{1}{2\pi j} \oint_{\Gamma} \frac{H_{ij}^{(1)}(z)}{z-1} z^k dz + (1-\lambda) \frac{1}{2\pi j} \oint_{\Gamma} \frac{H_{ij}^{(2)}(z)}{z-1} z^k dz \\ &= \lambda \Phi_k(H^{(1)}) + (1-\lambda) \Phi_k(H^{(2)}) \end{aligned} \quad (4.18)$$

for $k = 0, 1, \dots, N$ also satisfy Eq. 4.16.

In the above Eq. 4.18, $H^{(1)}$ and $H^{(2)}$ are in $\mathcal{H}_{stable}$ and $0 \leq \lambda \leq 1$. An important conclusion can be drawn here that since the performance specifications for time domain quantities correspond to sets of acceptable closed-loop transfer matrices, if the time domain quantities under constraints are *convex* in nature (e.g., it is easy to verify that the set $\{r(k) \mid r_{min}(k) \leq r(k) \leq r_{max}(k) \text{ for } k = 0, 1, 2, \dots, N\}$ is convex), and they are able to be described in Z domain, then the Z domain descriptions are *convex functions* of the transfer matrix H . The *convexity* is preserved between the two different domains because the Z – *transformation* and its inverse are linear operations. This idea is used in the next section.

4.2.3 Bounded Quantities in Time Domain

Very often the control designer has to deal with performance specifications related to bounded quantities in the time domain. For instance, the inputs of the closed-loop transfer functions H_{ip} , H_{iq} , H_{ir} as entries of H all affect the i th regulated output z_i of the system. Suppose z_i is the actuator effort and must be bounded within the range $\pm b$, and the p, q, r th inputs (which represent some command, noise or disturbance) w_p , w_q , w_r are also bounded by $\pm b_p$, $\pm b_q$, $\pm b_r$, respectively. Specifications of this type can be described by the L_1 constraints:

$$b_p \int_0^\infty |h_{ip}(t)| dt + b_q \int_0^\infty |h_{iq}(t)| dt + b_r \int_0^\infty |h_{ir}(t)| dt \leq b,$$

or in discretized form,

$$b_p \sum_{k=0}^N |h_{ip}(k)| + b_q \sum_{k=0}^N |h_{iq}(k)| + b_r \sum_{k=0}^N |h_{ir}(k)| \leq b \quad (4.19)$$

where, h_{ip} , h_{iq} and h_{ir} are the impulse responses of H_{ip} , H_{iq} and H_{ir} .

Since the triple $\langle h_{ip}, h_{iq}, h_{ir} \rangle$ is an element of the *convex set* $\{ \langle h_{ip}, h_{iq}, h_{ir} \rangle \mid \text{all } \langle \cdot \rangle \text{ that satisfy the constraint (4.19)} \}$, we can conclude that if the constraint 4.19 can be described properly (we will see in Section 4.4), then the descriptions are *convex functions* associated with the transfer matrix H .

4.2.4 Bounds in Frequency Domain

Classical frequency domain specifications regarding bandwidth or peaking can be expressed by

$$|H_{ij}(e^{j\Omega})| \leq b(\Omega), \text{ for } |\Omega| \leq \Omega_b \quad (4.20)$$

where $b(\Omega)$ is some bound function, Ω_b is the bandwidth. Constraint 4.20 may mean the specification of disturbance rejection over the given band Ω_b . Let

$$\Phi(H) = |H_{ij}(e^{j\Omega})|, \quad (4.21)$$

it is obviously a *convex function* since for any $H^{(1)}$ and $H^{(2)}$ in $\mathcal{H}_{stable}$,

$$\begin{aligned} \Phi(\lambda H^{(1)} + (1 - \lambda)H^{(2)}) &= |\lambda H_{ij}^{(1)}(e^{j\Omega}) + (1 - \lambda)H_{ij}^{(2)}(e^{j\Omega})| \\ &\leq \lambda |H_{ij}^{(1)}(e^{j\Omega})| + (1 - \lambda) |H_{ij}^{(2)}(e^{j\Omega})| \\ &= \lambda \Phi(H^{(1)}) + (1 - \lambda) \Phi(H^{(2)}) \end{aligned} \quad (4.22)$$

also satisfies Eq. 4.20.

Many robust considerations are the topics of H_∞ control theory and therefore relate to bounds on the “infinity norm”, as defined by Eq. A.9 for the scalar case or Eq. A.10 in general. For a typical closed-loop system shown in Fig. 4.3,

a classical measure of the robustness with respect to “loop perturbations”, is the stability margin that is described by the *M-circle radius*, the minimum distance between the point $(-1, j0)$ and the loop gain PK on the Nyquist plot, as illustrated by Fig. 4.4,

Figure 4.3: A typical closed-loop system.

Since

$$\begin{aligned}
 M &= \min_{\Omega} \text{dist}(-1, PK) \\
 &= \min_{\Omega} |I + P(e^{j\Omega})K(e^{j\Omega})| \\
 &= \frac{1}{\max_{\Omega} |(I + P(e^{j\Omega})K(e^{j\Omega}))^{-1}|} \\
 &= \frac{1}{\|H_0\|_{\infty}},
 \end{aligned}$$

where $\|H_0\|_{\infty} \triangleq (I + P(e^{j\Omega})K(e^{j\Omega}))^{-1}$, we have

$$\|H_0\|_{\infty} = \frac{1}{M - \text{circle radius}} \quad (4.23)$$

that is, proper bounds on $\|H_0\|_{\infty}$ will guarantee certain stability margin of the closed-loop system, maximum stability margin can be achieved by minimizing $\|H_0\|_{\infty}$.

Figure 4.4: Illustration of M-circle radius.

For generality, here we consider P and K are matrices and $\|H_0\|_\infty$ is defined according to Eq. A.10. We can apply the general idea to specific loops in a *MIMO* system, for instance, the example given in Section 3.1.3 has a closed-loop transfer matrix shown in Eq. 3.4, if we set a proper bound on the infinity-norm of the transfer function, which is the 2,3 th entry of H , i.e.,

$$\|H_{23}\|_\infty = \left\| \frac{1}{1 + RK_2} \right\|_\infty \leq b,$$

then the stability margin of the entire transfer matrix will be guaranteed.

A major concern in robust control is the inaccuracy in plant modeling. Assuming that the actual plant $\tilde{P}$ can be expressed as

$$\tilde{P} = P + \Delta \tag{4.24}$$

where P is the *modeled* plant, Δ represents the *perturbation* which may include the unmodeled, neglected, or unknown dynamics. With the consideration of the existence of Δ , the closed-loop system shown in Fig. 4.3 should be redrawn as

Figure 4.5: Closed-loop system with plant perturbation.

Fig. 4.5 can be rearranged as a single loop with two blocks in it: one is the perturbation Δ , the other is what “sees” by Δ , as shown in Fig. 4.6.

According to Fig. 4.6, the perturbation Δ sees exactly the closed-loop transfer matrix without consideration of plant inaccuracy. Also noticed is that the rightmost part of Fig. 4.6 is exactly the same as Fig. A.3. With the assumption that Δ, P, K are all *LTI* and Δ, H are both stable, then from *small gain theorem* discussed in Section A.5, we have the conclusion that the control system

Figure 4.6: Equivalence of Fig. 4.5 seen by Δ .

with plant perturbation will be still stable as long as $\|\Delta\|_\infty \|H\|_\infty < 1$. Thus, if $\|H\|_\infty$ is bounded by $\frac{1}{l}$, the closed-loop system can tolerate a plant perturbation Δ up to $\|\Delta\|_\infty < l$. Again, smaller $\|H\|_\infty$ implies larger guaranteed stability margin.

For the purpose of generality, the above discussion is again based on *MIMO* systems. The perturbation matrix Δ is extracted from each loop of the system and the $\|\cdot\|_\infty$ is in the *maximum singular value* sense. We can obviously apply the idea to each individual loop by specifying the infinity norm bound on each closed-loop transfer function as an entry of the transfer matrix H .

For a *multiplicative* plant perturbation, we can easily turn it into an *additive* perturbation, by simply looking at Fig. 4.7:

Figure 4.7: Converting a multiplicative plant perturbation into an additive perturbation.

From Fig. 4.7, we have

$$\Delta' = (\Delta - I)P.$$

When both Δ and P are *LTI* and stable, Δ' is obviously *LTI* and stable. Thus the above discussion can be also applied to the case when *multiplicative* plant perturbation exists.

Finally, we need to show that the $\|H\|_\infty$ norm is a *convex function* of H .

Let

$$\Phi(H) = \|H\|_\infty \tag{4.25}$$

For any two transfer matrices $H^{(1)}$ and $H^{(2)}$ in $\mathcal{H}_{stable}$, if $\|H^{(1)}\|_\infty \leq b$ and

$\|H^{(2)}_{ij}\|_\infty \leq b$ (b is the bound) and $0 \leq \lambda \leq 1$,

$$\begin{aligned}
\Phi(\lambda H^{(1)} + (1 - \lambda)H^{(2)}) &= \|\lambda H^{(1)}_{ij} + (1 - \lambda)H^{(2)}_{ij}\|_\infty \\
&\leq \|\lambda H^{(1)}_{ij}\|_\infty + \|(1 - \lambda)H^{(2)}_{ij}\|_\infty \\
&= \lambda\|H^{(1)}_{ij}\|_\infty + (1 - \lambda)\|H^{(2)}_{ij}\|_\infty \\
&= \lambda\Phi(H^{(1)}) + (1 - \lambda)\Phi(H^{(2)}) \\
&\leq b
\end{aligned} \tag{4.26}$$

In deriving Eq. 4.26, we have used the properties of the *norm*.

Now we conclude this section with the understanding that a wide variety of performance specifications are *convex functions* of the closed-loop transfer matrix H . The reader may have noticed that some of the specifications such as Eq. 4.13 and Eq. 4.18 are not *strictly convex*, which means that the solution of convex programming may not be unique.

4.3 Parameterization of the Closed-loop Transfer Matrix and All Stabilizing Controllers

We have demonstrated that by express the closed-loop transfer matrix H_{zw} (simply denoted as H) in the free parameter representation form, $\mathcal{H}_{stable}$, the set of stable closed-loop transfer matrices is convex, and many performance specifications are convex functions on $\mathcal{H}_{stable}$. In this section we give details of this procedure which will also lead to the parameterization of all controllers

which stabilize H . For simplicity, in our discussion the generic symbol $\mathbf{S}_M$ is used to denote the *matrix ring* $\mathbf{S}^{m \times n}$ which is defined in Section A.1, since for much of the time, the actual dimensions of the matrices under discussion are not important to the concepts being put forward.

Rewriting the closed-loop transfer matrix:

$$H = P_{zw} - P_{zu}K(I + P_{yu}K)^{-1}P_{yw} \quad (4.27)$$

Since we assume the plant is open-loop stable, matrices P_{zw} , P_{zu} , P_{yu} and P_{yw} in Eq. 4.27 are all in $\mathbf{S}_M$. Thus, any controller which stabilizes P_{yu} also stabilizes H . Let

$$\mathcal{K} = \{K \mid K \text{ stabilizes } H\} \quad (4.28)$$

denote all *LTI* controllers that stabilize H . For an arbitrary nominal controller $K_{nom} \in \mathcal{K}$, according to discussions in Section A.2, since it stabilizes P_{yu} , K_{nom} and P_{yu} should satisfy the generalized Bezout Identity, therefore we have the following:

$$P_{yu} = ND^{-1} = \tilde{D}^{-1}\tilde{N} \quad (4.29)$$

$$K_{nom} = Y^{-1}X = \tilde{X}\tilde{Y}^{-1} \quad (4.30)$$

$$XN + YD = I \quad (4.31)$$

$$\tilde{N}\tilde{X} + \tilde{D}\tilde{Y} = I \quad (4.32)$$

where, N , D , $\tilde{N}$, $\tilde{D}$, X , Y , $\tilde{X}$, $\tilde{Y}$ are all in $\mathbf{S}_M$.

Based on the *arbitrary* nominal controller $K_{nom} \in \mathcal{K}$, the parameterization of H and $\mathcal{K}$ can be described by the following two facts:

Fact 1. (Parameterization of the closed-loop transfer matrix)

The closed-loop transfer matrix H in Eq. 4.27 can be parameterized as (rewrite Eq. 4.3)

$$H_{zw} = T_1 + T_2 Q T_3 \quad (4.33)$$

where matrix Q is the free parameter and it relates to the controller K by the following bilinear transformation:

$$\begin{aligned} K &= (Y - Q\tilde{N})^{-1}(X + Q\tilde{D}), |Y - Q\tilde{N}| \neq 0 \\ &= (\tilde{X} + DQ)(\tilde{Y} - NQ)^{-1}, |\tilde{Y} - NQ| \neq 0 \end{aligned} \quad (4.34)$$

Proof. The proof is straightforward. By substituting Eq. 4.34 into Eq. 4.27 and using Equations 4.29 through 4.32, we have

$$H = \underbrace{P_{zw} - P_{zu}DXP_{yw}}_{T_1} - \underbrace{P_{zu}D}_{T_2} Q \underbrace{\tilde{D}P_{yw}}_{T_3} \quad (4.35)$$

Note that in the above Eq. 4.35, T_1 , T_2 and T_3 are *fixed* since they only relate to the plant P and the known nominal controller K_{nom} , therefore Q is the only *free* parameter and is structurely *exposed*.

Fact 2. (Parameterization of all stabilizing controllers)

The set of all stabilizing LTI controllers, which is denoted by $\mathcal{K}$ in Eq. 4.28, can be parameterized as

$$\begin{aligned} \mathcal{K}(Q) &= \{(Y - Q\tilde{N})^{-1}(X + Q\tilde{D}) \mid Q \in \mathbf{S}_M, |Y - Q\tilde{N}| \neq 0\} \\ &= \{(\tilde{X} + DQ)(\tilde{Y} - NQ)^{-1} \mid Q \in \mathbf{S}_M, |\tilde{Y} - NQ| \neq 0\} \end{aligned} \quad (4.36)$$

Remarks. Eq. 4.36 states the following:

1. For any $Q \in \mathbf{S}_M$ with appropriate dimensions, the controller determined by Eq. 4.34 stabilizes P .
2. Conversely, for any *LTI* controller that stabilizes P , it can be described by Eq. 4.34 for some $Q \in \mathbf{S}_M$.

Proof.

1. This part is very easy, by letting

$$\begin{aligned} K &= (Y - Q\tilde{N})^{-1}(X + Q\tilde{D}) = \mathcal{Y}^{-1}\mathcal{X} \\ &= (\tilde{X} + DQ)(\tilde{Y} - NQ)^{-1} = \tilde{\mathcal{X}}\tilde{\mathcal{Y}}^{-1} \end{aligned}$$

and then check the *generalized Bezout Identity* (Eq. 4.31 and Eq. 4.32) for K and P_{yu} , or even simpler, by just looking at Eq. 4.35 and notice that all “fixed” matrices are in $\mathbf{S}_M$.

2. Since K_{nom} stabilizes P , the *generalized Bezout Identity* is satisfied so that Eq. 4.29 through Eq. 4.35 hold. Now suppose K is *any* controller that stabilizes P , then K and P_{yu} must satisfy the *generalized Bezout Identity* and therefore we have

$$K = \mathcal{Y}^{-1}\mathcal{X} = \tilde{\mathcal{X}}\tilde{\mathcal{Y}}^{-1} \tag{4.37}$$

$$\mathcal{X}N + \mathcal{Y}D = I \tag{4.38}$$

and

$$\tilde{N}\tilde{\mathcal{X}} + \tilde{D}\tilde{\mathcal{Y}} = I \quad (4.39)$$

From Eq. 4.31 and Eq. 4.38, we have

$$(Y - \mathcal{Y})D = (\mathcal{X} - X)N. \quad (4.40)$$

From Eq. 4.32 and Eq. 4.39, we have

$$\tilde{D}(\tilde{Y} - \tilde{\mathcal{Y}}) = \tilde{N}(\tilde{\mathcal{X}} - \tilde{X}) \quad (4.41)$$

Let

$$Q = (\mathcal{X} - X)\tilde{D}^{-1} \quad (4.42)$$

yields

$$\begin{cases} \mathcal{X} &= X + Q\tilde{D} \\ \mathcal{Y} &= Y - Q\tilde{N} \end{cases} \quad (4.43)$$

Let

$$Q = D^{-1}(\tilde{\mathcal{X}} - \tilde{X}) \quad (4.44)$$

yields

$$\begin{cases} \tilde{\mathcal{X}} &= \tilde{X} + DQ \\ \tilde{\mathcal{Y}} &= \tilde{Y} - NQ \end{cases} \quad (4.45)$$

Note that Eq. 4.43 and Eq. 4.45 are consistent with Eq. 4.34.

To see Q determined by Eq. 4.42 and Eq. 4.44 are in $\mathbf{S}_M$, notice that from Eq. 4.31, we have

$$Y = (I - XN)D^{-1}$$

$$\begin{aligned}
&= D^{-1} - X(ND^{-1}) \\
&= D^{-1} - XP_{yu},
\end{aligned}$$

so that

$$D^{-1} = Y + XP_{yu}$$

stable because Y , X and P_{yu} are all in $\mathbf{S}_M$. Therefore Q determined by Eq. 4.44 is stable since $\tilde{X}$ and $\tilde{X}$ are in $\mathbf{S}_M$.

Similarly, by using Eq. 4.32, we have

$$\tilde{D}^{-1} = \tilde{Y} + P_{yu}\tilde{X}$$

stable and therefore Q determined by Eq. 4.42 is stable.

A general and more regorous proof of *Fact 2* can be found in [Vidyasagar 85] for the case P_{yu} not in $\mathbf{S}_M$.

As a conclusion of this section, *Fact 1* enables us to express the closed-loop transfer matrix H in a *free parametric representation* manner in which H depends on a single exposed parameter, the matrix Q so that the set $\mathcal{H}_{stable}$ is *convex* with respect to Q ; *Fact 2* tells us that by varing Q over the entire set $S^{n_{ctrl} \times n_{sens}}$, we can achieve *all* possible *LTI* controllers that stabilizes the plant P , thus the controller design can be done by searching Q over the entire $S^{n_{ctrl} \times n_{sens}}$ subject to the given performance specifications.

A technical problem related to this section: how to find the left and right comprime factorizations for the plant P and the nominal controller K_{nom} , will

be explained in Chapter 6.

4.4 Finite Dimensional Approximation of the Solution Space

As outlined by Section 4.1, the key of this design method is the *convex programming* approach. Since there is no known analytic method that can handle the multiobjective, multiconstraint problem described by different types of performance specifications, a numerical approach becomes a natural choice. However, insofar the property of *convexity* is described in an *infinity dimensional vector space*, in order to make the numerical approach feasible, it is necessary to redescribe our problem in a *finite dimensional Euclidean Space*. This is probably the toughest problem, but was handled here via a simple minded approach: descretizing and trunction.

For the parameterized representation of the closed-loop transfer matrix $H = T_1 + T_2QT_3$, we use H_{ij} , $T_{ij}^{(1)}$, $T_{ij}^{(2)}$, Q_{ij} , $T_{ij}^{(3)}$ to represent the i, j th entry of the corresponding matrix, then

$$H_{ij} = T_{ij}^{(1)} + \sum_{l=1}^n T_{il}^{(2)} \sum_{k=1}^m Q_{lk} T_{kj}^{(3)} \quad (4.46)$$

To avoid complicated notations in our discussion, the concepts are explained for *SISO* case, i.e., in the equation

$$H = T_1 + T_2QT_3 \quad (4.47)$$

H , T_1 , Q , T_2 and T_3 are all degenerated as transfer functions in $\mathbf{S}(z)$. The idea is valid for *MIMO* cases by noticing that the scalar equation 4.47 can be seen as one term in the generalized representation Eq. 4.46.

Our main approach is to approximate T_1 , Q , T_2 and T_3 in $\mathbf{S}(z)$ by stable *FIR* filters with *finite* number of delays (this is done through truncation), i.e.,

$$\left\{ \begin{array}{l} T_1 \approx \sum_{i=0}^n c_{1i} z^{-i} \\ T_2 \approx \sum_{i=0}^n c_{2i} z^{-i} \\ T_3 \approx \sum_{i=0}^n c_{3i} z^{-i} \\ Q \approx \sum_{i=0}^n x_i z^{-i} \end{array} \right. \quad (4.48)$$

Note that in the above Eq. 4.48, T_1 , T_2 and T_3 are *fixed* and therefor the coefficients c_{1i} , c_{2i} , c_{3i} for $i = 1, 2, \dots, n$ are *constants*, while Q is the *free* parameter to be determined, so the x_i 's become *decision variables* to be solved by the convex programming problem.

For steady state performance specifications, from Eq. 4.12 we have

$$\begin{aligned} \Phi(H) &= H(z)|_{z=1} = \tilde{\Phi}(Q) = T_1(z)|_{z=1} + T_2(z)Q(z)T_3(z)|_{z=1} \\ &\approx \sum_{i=0}^n c_{1i} z^{-i}|_{z=1} + (\sum_{i=0}^n c_{2i} z^{-i})(\sum_{i=0}^n x_i z^{-i})(\sum_{i=0}^n c_{3i} z^{-i})|_{z=1} \\ &= c_1 + c_2 \sum_{i=0}^n x_i = \hat{\Phi}(\mathbf{x}) = c \end{aligned} \quad (4.49)$$

where, $c_1 = \sum_{i=0}^n c_{1i}$, $c_2 = (\sum_{i=0}^n c_{2i})(\sum_{i=0}^n c_{3i})$ and c is a constant. The constraint function in Eq. 4.12 now becomes

$$\hat{\Phi}_0(\mathbf{x}) = \hat{c} \quad (4.50)$$

where, $\hat{c} = c - c_1$, and $\hat{\Phi}_0(\mathbf{x}) = c_2 \sum_{i=0}^n x_i$ is in n -dimensional Euclidean Space.

For transient reponse requirements specified in time domain, in Section 4.2.2 we showed that by discretizing, the specifications can be converted to a finite number of *convex constraints* that corresponding to each sampling moment, see Constraint 4.15. We now use step respons as an example, since

$$h(k) = t_1(k) + \sum_{j=0}^k \left(\sum_{i=0}^{l-1} x_i t_3(j-i) \right) t_2(k-j),$$

we have

$$\begin{aligned} s(n) &= \sum_{k=0}^n (t_1(k) + \sum_{j=0}^k (\sum_{i=0}^{l-1} x_i t_3(j-i)) t_2(k-j)) \\ &= \sum_{k=0}^n t_1(k) + \sum_{i=0}^{l-1} x_i (\sum_{k=0}^n \sum_{j=0}^k t_3(j-i) t_2(k-j)) \\ &= c_1 + c_2 \sum_{i=0}^{l-1} x_i = \hat{\Phi}_n(\mathbf{x}) \end{aligned} \quad (4.51)$$

where $c_1 = \sum_{k=0}^n t_1(k)$, $c_2 = \sum_{k=0}^n \sum_{j=0}^k t_3(j-i) t_2(k-j)$. The constraint function in Eq. 4.15 now becomes

$$s_{min}(i) \leq \hat{\Phi}_i(x) \leq s_{max}(i) \quad (4.52)$$

where, $s_{min}(n) = r_{min}(n) - c_1$, $s_{max}(n) = r_{max}(n) - c_1$, and $\hat{\Phi}_n(\mathbf{x}) = c_2 \sum_{i=0}^{l-1} x_i$ is in l -dimensional Euclidean Space.

The constraints generated by time domain boundary specifications are infinit dimensional. For their discretized form Constraint 4.19, if we divide the bound b with respect to each impulse response h at each sampling moment k , and notice that $|h(k)| \leq b_i$, then the L_1 type constraints can be handled in the same way as we did for transient response constraints.

The approach been discussed above applies to frequency domain boundary specifications in a similar way. Notice that the frequency domain boundary described by Inequality 4.20 or by specifications on $\|H\|_\infty$ norm (Eq. 4.25) are both infinite dimensional constraints. However, in our *FIR* approximations (Eq. 4.48), let $z = e^{j\Omega}$ and then discretize Ω on the unit circle, we can obtain a finite number of constraints with respect to each angle Ω_k . Note that for each specific angle $\Omega = \Omega_k$, z^{-i} is a constant value, the constraints are only depend on the decision variables x_i 's. This situation is very similar to the steady state constraints discussed before, except that the resulting constraint (or objective) functions here are in *complex* form, for example,

$$|A(\mathbf{x})| + j|B(\mathbf{x})| \leq c, \text{ for } \Omega = \Omega_k,$$

we have to further convert it into the form

$$c_1 \leq A^2(\mathbf{x}) + B^2(\mathbf{x}) \leq c_2, \text{ for } \Omega = \Omega_k,$$

again the function $\Phi_k(\mathbf{x}) = A^2(\mathbf{x}) + B^2(\mathbf{x})$ is in finite dimensional Euclidean Space.

To this point, we have discussed the controller design method on both conceptual and technical level, in the next chapter, we will show this approach can be really implemented.

Chapter 5

A Design Example

In Chapter 3, we introduced a control system configuration that the convex programming controller design approach relies on. In Section 3.1.3, a simple force control system introduced by [An and Hollerbach 87a] has been used as an example to show that how a control system can be represented by the aforementioned configuration. Moreover, modeling of a time delay in the feedback loop was discussed in Section 3.2. In this chapter, we continue to use the same example to demonstrate the controller design procedure.

5.1 The System

The force feedback control system was described in Subsection 3.1.3. Its modified version (with feedback delay) can be found in Fig. 3.12, we redraw the detailed schematic as below:

In Figure 5.1, R is the dynamic model of the coupled robot-environment system; the perturbed plant matrix $\tilde{P}$ is (rewrite Eq.3.5):

Figure 5.1: The force control system with feedback delay.

$$\tilde{P} = \begin{bmatrix} P_{zw} & \tilde{P}_{zu} \\ P_{yw} & \tilde{P}_{yu} \end{bmatrix} = \begin{bmatrix} R & R & 0 & \tilde{R} \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ R & R & 1 & \tilde{R} \end{bmatrix} \quad (5.1)$$

where $\tilde{R}$ is the perturbed robot-environment dynamics due to the feedback delay. We now assume that the simplified robot-environment system can be modeled by a mass-spring-damper system shown in Fig. 2.1. With known parameter values, R and $\tilde{R}$ in Eq. 5.1 can be determined explicitly via approaches

explained in Subsection 3.2.2. For instance, with the following parameter values:

$$m_r = 1 \text{ kg},$$

$$b_e = 0.1 \text{ kg/sec},$$

$$k_e = 60,000 \text{ N/m},$$

$$f_s = 200 \text{ Hz},$$

$$\tau = 0.6 T,$$

where f_s is the sampling frequency, T is the sampling period and τ is the feedback delay time. The discrete models for R and $\tilde{R}$ are

$$R = \frac{0.6607z + 0.6606}{z^2 - 0.6782z + 0.9995} \quad (5.2)$$

and

$$\tilde{R} = \frac{0.1176z^2 + 0.9457z + 0.2580}{z^3 - 0.6782z^2 + 0.9995z}$$

respectively. In the design example we will give later on, the delay τ is considered equal to T , the sampling period. In that case, $\tilde{R}$ is

$$\tilde{R} = \frac{0.6607z + 0.6606}{z^3 - 0.6782z^2 + 0.9995z} \quad (5.3)$$

Both R and $\tilde{R}$ reflect the effect of zero-order-hold. $\tilde{R}$ has also considered the effect of feedback delay. Note that by discretizing, the perturbed plant $\tilde{P}$ remains as *LTI*. Further, if P is open-loop stable, then $\tilde{P}$ is also open-loop stable, i.e., P_{zw} , P_{yu} , $\tilde{P}_{zu}$ and $\tilde{P}_{yu}$ are all in S_M .

5.2 Stable Coprime Factorizations

As discussed in Chapter 4, before we can start the convex programming procedure in controller design, a necessary step is to represent the closed-loop transfer matrix H_{zw} in the *free parameterization* form as shown in Eq. 4.35. In order to do so, we must first complete the procedure of *stable coprime factorization* for both $\tilde{P}_{yu}$ and K_{nom} .

Represent the perturbed open-loop transfer matrix $\tilde{P}_{yu}$ (from the input signal vector u to the measured signal vector y) in its state-space form

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Eu(k), \end{aligned}$$

if it is both controllable and observable, then there are some controller K_{cop} and observer L_{cop} such that $A - BK_{cop}$ and $A - L_{cop}C$ are stable. Moreover, we will have the following (see [Vidyasagar 85]):

$$\begin{aligned} \tilde{N} &= C(zI - A + L_{cop}C)^{-1}(B - L_{cop}E) + E \in \mathbf{S}_M, \\ \tilde{D} &= I - C(zI - A + L_{cop}C)^{-1}L_{cop} \in \mathbf{S}_M, \\ N &= (C - EK_{cop})(zI - A + BK_{cop})^{-1}B + E \in \mathbf{S}_M, \\ D &= I - K_{cop}(zI - A + BK_{cop})^{-1}B \in \mathbf{S}_M, \\ X_{temp} &= K_{cop}(zI - A + L_{cop}C)^{-1}L_{cop} \in \mathbf{S}_M, \\ Y_{temp} &= I + K_{cop}(zI - A + L_{cop}C)^{-1}(B - L_{cop}E) \in \mathbf{S}_M, \end{aligned}$$

$$\tilde{X}_{temp} = K_{cop}(zI - A + BK_{cop})^{-1}L_{cop} \in \mathbf{S}_M,$$

$$\tilde{Y}_{temp} = I + (C - EK_{cop})(zI - A + BK_{cop})^{-1}L_{cop} \in \mathbf{S}_M,$$

$$\tilde{P}_{yu} = ND^{-1} = \tilde{D}^{-1}\tilde{N},$$

$$K_{temp} = Y_{temp}^{-1}X_{temp} = \tilde{X}_{temp}\tilde{Y}_{temp}^{-1}$$

and

$$\begin{bmatrix} Y_{temp} & X_{temp} \\ -\tilde{N} & \tilde{D} \end{bmatrix} \begin{bmatrix} D & -\tilde{X}_{temp} \\ N & \tilde{Y}_{temp} \end{bmatrix} = I.$$

in other words, K_{temp} stabilizes $\tilde{P}_{yu}$.

In our implementation, K_{cop} , L_{cop} and K_{nom} are independently chosen therefore usually $K_{nom} \neq K_{temp}$. Eq. 4.35 shows that the *free parameterization* of H_{zw} requires the stable coprime factorization of K_{nom} , i.e., to find $X \in \mathbf{S}_M$ and $Y \in \mathbf{S}_M$ such that $K_{nom} = Y^{-1}X$ and $XN + YD = I$. By *Factor 2* in Section 4.3, if K_{temp} is known as a stabilizing controller and its stable coprime factorization X_{temp} and Y_{temp} are obtained, then there exists some $Q_i \in \mathbf{S}_M$ such that

$$K_{nom} = (Y_{temp} - Q_i\tilde{N})^{-1}(X_{temp} + Q_i\tilde{D}) \quad (5.4)$$

then with a chosen K_{nom} , Q_i can be determined from the above Eq. 5.4:

$$Q_i = (Y_{temp}K_{nom} - X_{temp})(\tilde{D} + \tilde{N}K_{nom})^{-1} \in \mathbf{S}_M \quad (5.5)$$

and the stable coprime factorization of K_{nom} can be found as

$$X = X_{temp} + Q_i\tilde{D}$$

$$Y = Y_{temp} - Q_i \tilde{N}$$

and $X \in \mathbf{S}_M$, $Y \in \mathbf{S}_M$, $XN + YD = I$. By Eq. 4.35, once the stable coprime factorization of $\tilde{P}_{yu}$ and K_{nom} have been found, the “fixed” parameters T_1 , T_2 and T_3 can be calculated:

$$T_1 = P_{zw} - \tilde{P}_{zu}DXP_{yw},$$

$$T_2 = -\tilde{P}_{zu}D,$$

$$T_3 = \tilde{D}P_{yw}$$

where X is obtained from the stable coprime factorization of K_{nom} , D and $\tilde{D}$ are obtained from the stable coprime factorization of $\tilde{P}_{yu}$.

As discussed in Section 4.4, in order to convert the design problem into a finite dimensional Euclidean Space, T_1 , T_2 and T_3 need to be further discretized and truncated.

5.3 Choice of Nominal Controller

Theoretically, the only restriction on the nominal controller K_{nom} is to be *LTI* and to stabilize $\tilde{P}_{yu}$, so the simplest form of K_{nom} is a constant, i.e., a proportional controller.

Discussion in Chapter 2 reveals that the coupled robot-stiff environment is a highly underdamped system, and with a time delay in the force feedback action, negative force feedback tends to destabilize the system while positive

force feedback tends to stabilize the system. These results were obtained based on the analysis for continuous systems, our software experiments show that they are also valid for the discretized system which has also considered the zero-order-hold effect. Fig. 5.2 shows the positions of the open-loop poles, Fig. 5.3 shows the step response of the open-loop system (the mass-spring-damper model with $m_r = 1\text{ kg}$, $b_e = 0.1\text{ kg/sec}$, $K_e = 60,000\text{ N/m}$ and $\tau = T$), Fig. 5.4 shows the root-locus of the closed-loop system when delayed negative force feedback is applied, Fig. 5.5 shows that a slight negative feedback will easily drive the underdamped system unstable, Fig. 5.6 shows the root-locus of the closed-loop system when delayed positive force feedback is applied, Fig. 5.7 shows that the delayed positive feedback significantly reduced the oscillation and improved stability of the system.

Although theoretically different K_{nom} should make no difference to the final results as long as they all stabilize $\tilde{P}_{yu}$, practically, different choices of K_{nom} can make big difference. In Chapter 4 we have explained that in order to cast many of the performance specifications into convex constraints, our basic approach is to discretize the specifications either on the time axis or on the unit circle. In this way, one continuous performance specification will be converted to a large number of convex constraints (or objective functions) with each corresponding to a specific sampling point (time or angle). It is imaginable that a poor choice of K_{nom} will cause the convex programming problem to start

Figure 5.2: Positions of the open-loop poles.

from an undesirable position. For example, if the performance specification is a set of bounds on the step response described by Eq. 4.14 and Fig. 4.2, a K_{nom} that barely stabilizes $\tilde{P}_{yu}$ may correspond to a trajectory of step response which is far beyond the given bounds, as shown in Fig. 5.8.

Fig. 5.8 shows that the convex programming starts from a position in which almost all constraints are violated. In our practice, this not only significantly increased the computational time in the convex programming but also frequently led to failure, since the convex programming software we have used has set limitations on the number of iterations. Therefore, in experiments, we always choose a *positive feedback* proportional nominal controller.

Figure 5.3: Step response of the open-loop system.

5.4 Design Tools

Our design procedure mainly contains the pre/post convex programming part and the convex programming part.

The pre/post convex programming is done by using MATLAB. This part includes the following tasks:

- model the system with delay in feedback actions
- choose the nominal controller K_{nom}
- find the stable coprime factorization for $\tilde{P}_{yu}$ and K_{nom}
- calculate, discretize and truncate T_1 , T_2 and T_3

Figure 5.4: Root-locus of the closed-loop system with delayed negative force feedback ($0 \leq K \leq 1$).

- find the initial Q from the nominal controller K_{nom} before the convex programming
- find the desired controller K from the optimal Q after the convex programming

All these procedures have been described in the previous discussions. In our simulation, the nominal controller K_{nom} was selected interactively at run time. with guidelines as Fig. 5.6, it is very easy to find a suitable K_{nom} to start the convex programming; the stable controller K_{cop} and observer L_{cop} for the stable coprime factorization of $\tilde{P}_{yu}$ were found by using the MATLAB function *place*; entries of Q were approximated by *FIR* filters with 15 delays, the total

Figure 5.5: Step-response of the closed-loop system with delayed negative force feedback ($K = 0.01$).

number of decision variables is thus 30 (since $Q \in \mathbf{S}^{1 \times 2}$ in this example); the finite dimensional approximation of T_1 , T_2 and T_3 are done by using the MATLAB function *dimpulse*.

The implementation of the convex programming is made possible by using QDES, a discrete time linear control system design specification compiler and solver which is developed in the Information Systems Laboratory at the Electrical Engineering Department, Stanford University. The input of the QDES compiler is a list of constraints or objective functions and data files generated by tools such as MATLAB. The constraints and objective functions are specified by a C-like *control specification language*. The output of the QDES

Figure 5.6: Root-locus of the closed-loop system with delayed positive force feedback ($-1 \leq K \leq 0$).

compiler is a convex program for the decision variables which can be used by a convex program solver. The solver attempts to solve the convex program starting from the initial values of the decision variables (i.e., the coefficients of the truncated initial Q which corresponds to K_{nom} , see Eq. 4.48). If the solver finds an optimal solution $\mathbf{x}^*$, then Q corresponding to the desired controller K is found and thus K can be determined by Eq. 4.34. On the other hand, if the solver finds the program to be infeasible, or a certain number of iterations has been reached without finding the optimal solution, the convex programming procedure will be discontinued.

Figure 5.7: Step response of the closed-loop system with delayed positive force feedback ($K = -0.1$).

5.5 Some Results

In this section, we show three groups of design specifications for the system in the example, and the simulation results obtained by using controllers corresponding to each design. To ease the reading, instead of using the *control specification language*, we present the design specifications in a straightforward way. The real implementation was illustrated in Section 4.4 and was handled by the QDES compiler. For each design, the convex programming starts at the same initial Q which corresponds to $K_{nom} = [0.1 \quad -0.1]$ (positive feedback), its endpoint force step response is shown in Fig. 5.7, its actuator effort is shown in Fig. 5.9.

Figure 5.8: The specification envelopes and the step response corresponding to a barely stabilizing nominal controller K_{nom} .

The closed-loop transfer matrix H_{zw} for the system in the given example is shown by Eq. 3.6. To visualize the relationship between each specification and each entry of H_{zw} , we detail the closed-loop input-output equation $z = H_{zw}w$ as below (refer to Fig. 5.1):

$$\begin{bmatrix} f_r \\ u_a \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \end{bmatrix} \begin{bmatrix} f_d \\ n_r \\ n_s \end{bmatrix} \quad (5.6)$$

Design 1 Specifications:

1. The unit step response of the endpoint force f_r to the desired contact

Figure 5.9: The actuator effort response to a unit step force command with the nominal controller.

force f_d should be inside the specified envelopes:

$$1 - 0.8e^{-14kT} \leq f_r(kT) \leq 1 + e^{-12kT},$$

for

$$k = 0, 1, \dots, 800$$

and

$$T = 0.005 \text{ second.}$$

This is a constraint on H_{11} .

2. To restrict the response of the endpoint force f_r to the force sensor noise

n_s by minimizing $\|H_{13}(e^{j\Omega})\|_\infty$ over $\Omega = [0, 2\pi)$.

The controller was obtained after the convex programming. The simulation results corresponding to this controller are shown in Figures 5.10 to 5.12. Figures 5.13 and 5.14 compare the endpoint force and the actuator step responses obtained by using the Design 1 controller to those responses obtained by using the nominal controller.

Figure 5.10: The endpoint force response to a unit step force command with the Design 1 controller.

Figure 5.13 shows that by using the Design 1 controller, the response speed of the endpoint force has been improved. However, Figure 5.14 shows that the actuator effort has been degraded. Both Figures 5.13 and 5.14 show that with the Design 1 controller, the endpoint force and the actuator effort step responses have large overshoots.

Figure 5.11: The actuator effort response to a unit step force command with the Design 1 controller.

Design 2 Specifications:

1. The endpoint force step response is similar to the one in Design 1 but the constraint is tightened:

$$1 - 0.8e^{-20kT} \leq f_r(kT) \leq 1 + e^{-18kT},$$

for

$$k = 0, 1, \dots, 800$$

and

$$T = 0.005 \text{ second.}$$

2. To minimize $\|H_{13}(e^{j\Omega})\|_{\infty}$ over $\Omega = [0, 2\pi)$, same as in Design 1.

Figure 5.12: The frequency response of H_{13} with the Design 1 controller.

3. To restrict the overshoot of the endpoint force. This is another constraint on H_{11} .
4. To restrict the actuator effort by minimizing its maximum absolute value in response to the force command f_d . This is a constraint on H_{21} .

Simulation results based on the controller obtained from Design 2 are shown in Figures 5.15 through 5.17.

Compare Figures 5.15 through 5.17 with Figures 5.10 through 5.12, the frequency response of H_{13} does not change a lot, while the improvement on endpoint force and actuator responses is significant (for comparison, the envelopes in Figure 5.15 and Figure 5.10 are same). Figures 5.18 and 5.19 compare the endpoint force and the actuator step responses obtained by

Figure 5.13: The endpoint force step responses (solid line: Design 1 controller; dotted line: nominal controller).

using the Design 2 controller to those responses obtained by using the nominal controller. They show that with the Design 2 controller, the system responds much faster without overshooting the endpoint force and the actuator effort. This could be very important to certain tasks such as robot fine grinding, and to many applications where the actuator efforts are bounded.

Design 3 Specifications: The set of design specifications are the same as those in Design 2 except the envelopes for the endpoint force step response are further tightened:

$$1 - 0.8e^{-22kT} \leq f_r(kT) \leq 1 + e^{-20kT},$$

Figure 5.14: The actuator effort step responses (solid line: Design 1 controller; dotted line: nominal controller).

for

$$k = 0, 1, \dots, 800$$

and

$$T = 0.005 \text{ second.}$$

The simulation results based on the obtained controller are shown in Figures 5.20 through 5.22.

Compare to the previous results, the Design 3 controller slightly further improved the system's response speed. However, a slight overshoot (1%) in the endpoint force was appeared, as Figure 5.20 shows.

Figure 5.15: The endpoint force response to a unit step force command with the Design 2 controller.

Figures 5.23 and 5.24 compare the endpoint force and the actuator step responses obtained by using the Design 3 controller to those responses obtained by using the nominal controller. The frequency response of H_{13} corresponding to the three designed controllers look quite similar since minimizing $\|H_{13}\|_{\infty}$ was required by all three designs. The result of this specification is, the frequency responses of H_{13} are flattened, in other words, the peak of H_{13} been suppressed. Figure 5.25 shows the frequency response of H_{13} corresponding to the nominal controller.

Figure 5.16: The actuator effort response to a unit step force command with the Design 2 controller.

Compare Figures 5.12, 5.17 and 5.22 to Figure 5.25, the designed controllers slightly reduced the peak value of $|H_{13}(e^{j\Omega})|$ by sacrificing the immunity to the force sensor noise at low frequency.

5.6 The Controller

This controller design method usually yields high-order controllers and the *balanced model reduction* technique [Moore 81, Boyd 86] has been applied in our experiments. As an example, the controller obtained from Design 3 has an order of 26, after the model reduction the new controller's order reduced to 15 while keeping the system's performance indistinguishable from the one

Figure 5.17: The frequency response of H_{13} with the Design 2 controller.

using the original controller. Figures 5.26 through 5.29 show that the frequency responses of the reduced-order controller match perfectly with those of the original 26 th order controller (note that the controller has the form of $[K_1(z) \ K_2(z)]$).

Table 5.1 lists the coefficients of the reduced-order controller.

The poles of the reduced-order controller and the closed-loop transfer functions in H_{zw} are plotted in Fig. 5.30 and Fig. 5.31, respectively. Note that in the controller matrix, $K_1(z)$ and $K_2(z)$ have the same poles; and all closed-loop transfer functions in H_{zw} have the same poles (refer to Eq. 3.6).

Table 5.2 lists the poles and zeros of the reduced-order controller.

Figure 5.18: The endpoint force step responses (solid line: Design 2 controller; dotted line: nominal controller).

Figure 5.19: The actuator effort step responses (solid line: Design 2 controller; dotted line: nominal controller).

Figure 5.20: The endpoint force response to a unit step force command with the Design 3 controller.

Figure 5.21: The actuator effort response to a unit step force command with the Design 3 controller.

Figure 5.22: The frequency response of H_{13} with the Design 3 controller.

Figure 5.23: The endpoint force step responses (solid line: Design 3 controller; dotted line: nominal controller).

Figure 5.24: The actuator effort step responses (solid line: Design 3 controller; dotted line: nominal controller).

Figure 5.25: The frequency response of H_{13} with the nominal controller.

Figure 5.26: The magnitude response of $K_1(e^{j\Omega})$ (solid line: original controller; dotted line: reduced-order controller).

Figure 5.27: The phase response of $K_1(e^{j\Omega})$ (solid line: original controller; dotted line: reduced-order controller).

Figure 5.28: The magnitude response of $K_2(e^{j\Omega})$ (solid line: original controller; dotted line: reduced-order controller).

Figure 5.29: The phase response of $K_2(e^{j\Omega})$ (solid line: original controller; dotted line: reduced-order controller).

denominator	$K_1(z)$ numerator	$K_2(z)$ numerator
1.0000	1.1495	-0.0729
-1.1465	-2.4194	0.2296
-0.2154	1.0068	-0.4697
-0.1947	0.3084	0.8072
0.7451	0.6679	-1.2164
0.4608	-0.3669	1.7174
-0.1705	-0.8007	-2.3233
-0.9121	-0.0143	3.0359
-0.0507	0.2705	-3.8286
0.4043	0.8740	4.7937
0.8480	-0.3438	-5.8725
-0.4676	-0.6001	5.7163
-0.4519	0.3608	-3.9353
0.2630	-0.5474	2.1733
-0.5461	0.4815	-0.8025
0.5304	0.0209	0.0000

Table 5.1: Coefficients of of the reduced-order controller.

Figure 5.30: The poles of the reduced-order controller.

Figure 5.31: The poles of the closed-loop transfer functions.

poles	$K_1(z)$ zeros	$K_2(z)$ zeros
-0.9102	-0.0414	-0.0001
-0.8988+0.4377i	-0.8765+0.2122i	-1.0648+0.7796i
-0.8988-0.4377i	-0.8765-0.2122i	-1.0648-0.7796i
-0.5833+0.8119i	-0.6513+0.6894i	-0.5431+1.2501i
-0.5833-0.8119i	-0.6513-0.6894i	-0.5431-1.2501i
-0.1397+0.9899i	-0.1985+0.8847i	0.1750+1.3689i
-0.1397-0.9899i	-0.1985-0.8847i	0.1750-1.3689i
0.0183+0.7739i	0.0230+0.7596i	0.8106+1.0404i
0.0183-0.7739i	0.0230-0.7596i	0.8106-1.0404i
0.7212+0.6904i	0.7552+0.6962i	0.0180+0.7742i
0.7212-0.6904i	0.7552-0.6962i	0.0180-0.7742i
0.9256+0.3681i	0.9746+0.3642i	1.0856+0.4732i
0.9256-0.3681i	0.9746-0.3642i	1.0856-0.4732i
0.9850+0.1216i	1.0465+0.1096i	1.0942+0.1556i
0.9850-0.1216i	1.0465-0.1096i	1.0942-0.1556i

Table 5.2: Poles and zeros of the reduced-order controller.

Chapter 6

Conclusions

6.1 Summary

Stability study on a simple force controlled robot arm against a stiff environment reveals that for a lightly damped system, time delay in the feedback action is a critical issue to the system's stability. If the system's natural frequency is high, the tolerable delay could be very small. Based on this understanding, the attention of this thesis then focused on the controller design for such systems. By explicitly including the feedback delay in a generalized framework for control systems, the designed controllers are robust with respect to the feedback delay. The design procedure is based on convex programming which runs off-line. The obtained controllers can effectively stabilize the lightly damped system and significantly improve its performance.

6.2 Main Features and Advantages

The main features and advantages of this controller design method are:

- The time delay in the feedback action has been taken into account, instead of by just saying that “assuming the sampling rate is high enough”.
- The sample-hold effect has been taken into account.
- Allowing the designer to start directly with a set of performance specifications.
- The convex programming is a global optimal approach. Theoretically, if it finds a solution, then the solution is optimal in the sense that it minimizes the specified objective function while satisfying all the constraints. On the other hand, if the convex programming cannot find a solution, then the set of design specifications is infeasible, i.e., no *LTI* controller can do the job. This is important since it gives an absolute limit on the performance of *any* linear controller.
- Except to be *LTI* (rational function of z), there is no specific constraint on the controller with respect to its order and structure.

6.3 Main Disadvantages

Main disadvantages of this controller design method are:

- It is impossible to specify the order and structure of the controller. Usually the order and complexity of the designed controllers are high, even

after applying model reduction techniques. However, a low order linear controller can be compared to the obtained “best” controller.

- The finite dimensional approximation of the infinite dimensional solution space causes the loss of feasible solutions, unless the Q and T_i 's are approximated by *FIR* filters with sufficient length.
- For each design, the delay time is a fixed value, meaning that a prior knowledge about the time needed for implementing the controller is required. One solution is to make a conservative (longer) estimation on the delay time before design, then purposely add in delay time when implementing the controller, if necessary.

6.4 Further Research

Hardware experiments on a suitable testbed will be very helpful. With the availability of hardware implementation of long *FIR* filters, the flexibility and usefulness of this controller design method could be substantially increased.

Appendix A

Related Mathematical and Control Concepts

In this appendix, we list some mathematical and control concepts that are related to the thesis.

A.1 The Ring Concept

A *ring* is a set of elements together with two operations, an *addition* and a *multiplication* defined on it. Let $\mathbf{K}$ be a *ring*, we have

$$p + q \in \mathbf{K} \text{ and } p \cdot q \in \mathbf{K}, \quad \forall p, q \in \mathbf{K}.$$

If in addition to the above axioms, we also have

$$pq = qp, \quad \forall p, q \in \mathbf{K},$$

then $\mathbf{K}$ is called a *commutative ring*.

The following sets are *commutative rings*:

$\mathbf{R}[z]$: Polynomials of complex variable z with coefficients in $\mathbf{R}$.

$\mathbf{R}(z)$: Rational functions of z with coefficients in $\mathbf{R}$.

$\mathbf{R}_p(z)$: Proper rational functions of z with coefficients in $\mathbf{R}$.

$\mathbf{R}_{p,o}(z)$: Strictly proper rational functions of z with coefficients in $\mathbf{R}$.

$\mathbf{S}(z)$: Stable proper rational functions of z with coefficients in $\mathbf{R}$.

The following matrices are *noncommutative rings*:

$\mathbf{R}^{m \times n}[z]$: m by n matrices with entries in $\mathbf{R}[z]$.

$\mathbf{R}^{m \times n}(z)$: m by n matrices with entries in $\mathbf{R}(z)$.

$\mathbf{R}_p^{m \times n}(z)$: m by n matrices with entries in $\mathbf{R}_p(z)$.

$\mathbf{R}_{p,o}^{m \times n}(z)$: m by n matrices with entries in $\mathbf{R}_{p,o}(z)$.

$\mathbf{S}^{m \times n}(z)$: m by n matrices with entries in $\mathbf{S}(z)$.

A.2 Matrix Coprimeness and the Bezout Identity

If $A \in \mathbf{K}^{p \times q}$, $B \in \mathbf{K}^{r \times q}$, $C \in \mathbf{K}^{q \times p}$, $D \in \mathbf{K}^{q \times r}$ are matrices with entries in a *commutative ring* $\mathbf{K}$, and the equation below holds:

$$CA + DB = I,$$

then we say that the matrices A and B are *right coprime* and that the matrices C and D are *left coprime*.

Let $P \in \mathbf{R}_{p,o}^{n_o \times n_i}(z)$, a *stable left coprime factorization* (S.L.C.F.) of P is

$$P = \tilde{D}^{-1} \tilde{N},$$

with $\tilde{D} \in \mathbf{S}^{n_o \times n_o}$ and $\tilde{N} \in \mathbf{S}^{n_o \times n_i}$, i.e., there are $\tilde{X} \in \mathbf{S}^{n_i \times n_o}$ and $\tilde{Y} \in \mathbf{S}^{n_o \times n_o}$

such that

$$\tilde{N}\tilde{X} + \tilde{D}\tilde{Y} = I.$$

Similarly, a *stable right coprime factorization (S.R.C.F.)* of P is $P = ND^{-1}$, with

$$XN + YD = I,$$

where $N \in \mathbf{S}^{n_o \times n_i}$, $D \in \mathbf{S}^{n_i \times n_i}$, $X \in \mathbf{S}^{n_i \times n_o}$, $Y \in \mathbf{S}^{n_i \times n_i}$.

The *S.L.C.F.* and *S.R.C.F.* of matrix $P \in \mathbf{R}_{p,o}^{n_o \times n_i}(z)$ can be described by the *generalized Bezout Identity*:

$$\begin{bmatrix} Y & X \\ -\tilde{N} & \tilde{D} \end{bmatrix} \begin{bmatrix} D & -\tilde{X} \\ N & \tilde{Y} \end{bmatrix} = I.$$

which means the following equations:

$$ND^{-1} = \tilde{D}^{-1}\tilde{N} \tag{A.1}$$

$$Y^{-1}X = \tilde{X}\tilde{Y}^{-1} \tag{A.2}$$

$$XN + YD = I \tag{A.3}$$

$$\tilde{N}\tilde{X} + \tilde{D}\tilde{Y} = I. \tag{A.4}$$

A very important theorem in linear control systems is [Vidyasagar 85]:

For a closed-loop control system, if P is the plant transfer matrix which has both S.L.C.F. and S.R.C.F.: $P = \tilde{D}^{-1}\tilde{N} = ND^{-1}$, and C is the controller transfer matrix which has both the S.L.C.F. and S.R.C.F.: $C = Y^{-1}X =$

$\tilde{X}\tilde{Y}^{-1}$, then C stabilizes P if and only if P, C satisfy the generalized Bezout Identity described by Equations (4.1) through (4.4).

A.3 Convexity — Sets, Functions and Programming Problems

A *convex set* is a collection of points such that if $x^{(1)}$ and $x^{(2)}$ are any two points in the collection, the line segment joining them is also in the collection.

If S denotes the convex set, it can be defined mathematically as follows:

If $x^{(1)}, x^{(2)} \in S$, then $x \in S$,

where,

$$x = \alpha x^{(1)} + (1 - \alpha)x^{(2)}, \text{ for all } 0 \leq \alpha \leq 1. \quad (\text{A.5})$$

A function $f(x)$ which is defined on some convex set S is said to be *convex function* if for any λ ($0 \leq \lambda \leq 1$) and any pair of points $x^{(1)}$ and $x^{(2)}$ in S ,

$$f(\lambda x^{(1)} + (1 - \lambda)x^{(2)}) \leq \lambda f(x^{(1)}) + (1 - \lambda)f(x^{(2)}) \quad (\text{A.6})$$

that is, if the segment joining the two points lies entirely above or on the graph of $f(x)$.

If for all λ ($0 < \lambda < 1$) and $x^{(1)} \neq x^{(2)} \in S$,

$$f(\lambda x^{(1)} + (1 - \lambda)x^{(2)}) < \lambda f(x^{(1)}) + (1 - \lambda)f(x^{(2)}) \quad (\text{A.7})$$

then $f(x)$ is a *strictly convex function*.

An important property of a *convex function* $f(x)$ on S is that, any local minimum of it is also a global minimum. Further, if $f(x)$ is *strictly convex*, the point at which $f(x)$ obtains its minimum is *unique*.

Any optimization problem stated in the form Minimize $f(x)$ subject to

$$g_j(x) \leq 0, \quad j = 1, 2, \dots, m. \quad (\text{A.8})$$

is called a *convex programming problem* provided the objective function $f(x)$ and the constraint functions $g_j(x)$ are all *convex*. For a convex programming problem, there will be no relative minima or saddle points, any local optimal solution is a global solution. Again, if $f(x)$ is *strictly convex*, the optimal solution is *unique*.

A.4 Concept of H_∞ and $\|H\|_\infty$

$\mathbf{H}_\infty$ denotes the set of complex functions which are analytic and bounded on the set $\mathbf{D} = \{z : 1 \leq |z| \leq r\}$ where $r \rightarrow \infty$. Clearly, $\mathbf{H}_\infty$ is a *ring* and the ring of stable proper rational functions $\mathbf{S}(z)$ is a *subring* of $\mathbf{H}_\infty$, i.e., $\mathbf{S}(z) \subset \mathbf{H}_\infty$.

In Chapter 3 we mentioned the need to measure the “size” of a transfer function in some appropriate sense. According to the *maximum principle*, any function in $\mathbf{H}_\infty$ will achieve its maximum on the boundary of $\mathbf{D}$. For a stable proper rational transfer function $H(z) \in \mathbf{H}_\infty$, since it vanishes on the outer

boundary ($|z| = r, r \rightarrow \infty$) of $\mathbf{D}$, we have

$$\sup_{\substack{1 \leq |z| \leq r \\ r \rightarrow \infty}} |H(z)| = \sup_{0 \leq \Omega \leq 2\pi} |H(e^{j\Omega})|,$$

that is, the maximum only occurs on the unit circle. We use the “infinity norm” $\|H\|_\infty$ to denote the *maximum* of a transfer function:

$$\|H\|_\infty \triangleq \sup_{0 \leq \Omega \leq 2\pi} |H(e^{j\Omega})| \quad (\text{A.9})$$

The above concept can be extend to allow $H(z)$ to be a transfer matrix. In such case, $\|H\|_\infty$ is defined as

$$\|H\|_\infty \triangleq \sup_{0 \leq \Omega \leq 2\pi} \sigma_{\max}(H(e^{j\Omega})) \quad (\text{A.10})$$

where $\sigma(H(e^{j\Omega}))$ denotes the *maximum singular value* of matrix $H(e^{j\Omega})$.

A straightforward interpretation of Eq. A.9 and Eq. A.10 is that $\|H\|_\infty$ norm is the *peak* of the frequency response of a transfer function (or matrix) $H(z)$, as illustrated by Fig. A.1.

Loosely, suppressing the $\|H\|_\infty$ norm of a transfer function (or matrix) $H(z)$ will reduce the risk of large responses to noises with unknown spectra.

Another good interpretation of $\|H\|_\infty$ norm is to see it as the RMS gain of a transfer function H . The reason is, for a system shown in Fig. A.2,

the spectrum densities of input and output are $S_x(e^{j\Omega})$ and $S_y(e^{j\Omega})$, respectively. Since $S_y(e^{j\Omega}) = |H(e^{j\Omega})|^2 S_x(e^{j\Omega})$, we have

$$E[y^2] = \frac{1}{2\pi} \int_0^{2\pi} S_y(e^{j\Omega}) d\Omega$$

Figure A.1: Illustration of $\|H\|_\infty$ norm.

$$\begin{aligned}
 &= \frac{1}{2\pi} \int_0^{2\pi} |H(e^{j\Omega})|^2 S_x(e^{j\Omega}) d\Omega \\
 &\leq \frac{1}{2\pi} \int_0^{2\pi} \left(\sup_{0 \leq \Omega \leq 2\pi} |H(e^{j\Omega})| \right)^2 S_x(e^{j\Omega}) d\Omega \\
 &= \|H\|_\infty^2 \cdot \frac{1}{2\pi} \int_0^{2\pi} S_x(e^{j\Omega}) d\Omega \\
 &= \|H\|_\infty^2 E[x^2],
 \end{aligned}$$

therefore, $\|H\|_\infty$ norm is the worst case RMS value of output with input power less than or equal to one.

If H is a transfer matrix, the reasoning should be rewritten as

$$\begin{aligned}
 E[Y^T Y] &= \frac{1}{2\pi} \int_0^{2\pi} \text{Tr} S_Y(e^{j\Omega}) d\Omega \\
 &= \frac{1}{2\pi} \int_0^{2\pi} \text{Tr}(H(e^{j\Omega}) S_X(e^{j\Omega}) H^*(e^{j\Omega})) d\Omega
 \end{aligned}$$

Figure A.2: A simple system for interpreting $\|H\|_\infty$ norm.

$$\begin{aligned}
 &= \frac{1}{2\pi} \int_0^{2\pi} \text{Tr}(S_X(e^{j\Omega}) H^*(e^{j\Omega}) H(e^{j\Omega})) d\Omega \\
 &\leq \frac{1}{2\pi} \int_0^{2\pi} (\text{Tr} S_X(e^{j\Omega})) \sigma_{\max}(H^*(e^{j\Omega}) H(e^{j\Omega})) d\Omega \\
 &= \|H\|_\infty^2 \cdot \frac{1}{2\pi} \int_0^{2\pi} S_X(e^{j\Omega}) d\Omega \\
 &= \|H\|_\infty^2 E[X^T X],
 \end{aligned}$$

where H^* is the complex conjugate transpose of H .

An important use of the $\|H\|_\infty$ norm concept is the *small gain theorem* which is introduced in the next section.

A.5 The Small Gain Theorem [Zames 66]

The *small gain theorem* can be roughly presented as follows:

If a perturbation Δ and the closed-loop transfer function H seen by Δ can be arranged in a loop shown in Fig. A.3, and if the “size” of Δ and H are not too large, then the loop is stable, hence the perturbed system is stable.

Figure A.3: The loop for explaining the small gain theorem.

The theory itself is general: the perturbation Δ can be nonlinear and time-varying, and there is no restriction on the method in measuring the “size” of Δ and H . However, if both Δ and H are *LTI* and stable, then the *small gain theorem* can be stated as:

The perturbed closed-loop system is stable if $\|\Delta\|_{\infty} \cdot \|H\|_{\infty} < 1$.

To this point, we have listed all mathematical or control concepts that are related to our controller design. Notations used in Chapters 4 and 5 follow definitions given here.

Bibliography

- [An, Atkeson and Hollerbach 88] C. H. An, C. G. Atkeson and J. M. Hollerbach, 1988. Model-Based Control of Robot Manipulator. *The MIT Press*. 1988.
- [An and Hollerbach 87a] C. H. An and J. M. Hollerbach, 1987. Dynamic Stability Issues in Force Control of Manipulation. *Proc. 1987 IEEE Int. Conf. Robotics and Automation*. pp. 890-896.
- [An and Hollerbach 87b] C. H. An and J. M. Hollerbach, 1987. Kinematic Stability Issues in Force Control of Manipulation. *Proc. 1987 IEEE Int. Conf. Robotics and Automation*. pp. 897-903.
- [Anderson and Spong 87] R. J. Anderson and M. W. Spong, 1987. Hybrid Impedance Control of Robotic Manipulators. *Proc. 1987 IEEE Int. Conf. Robotics and Automation*. pp. 1073-1080.
- [Anderson and Spong 89] R. J. Anderson and M. W. Spong, 1989 (May). Bilateral Control of Teleoperators with Time Delay. *IEEE Trans. Automatic Control*. Vol. 34, No. 5, 1989.
- [Åström and Wittenmark 84] K.J. Åström and B. Wittenmark, 1984. Computer Controlled Systems. *Prentice-Hall, Inc. Englewood Cliffs, NJ 07632*
- [Barker 52] R.H. Barker, 1952. The Pulse Transefer Function and Its Applications to Sampling Servo Systems. *Proc. IEE*, 99, pt. 4, 302-17.
- [Boyd 86] S.P. Boyd, 1986. EE364 Class Notes. *Department of Electrical Engineering, Stanford University*.

- [Boyd et al 88] S.P. Boyd, V. Balakrishnan, C.H. Barratt, N.M. Khraishi, X. Li, D.G. Meyer and S.A. Norman, 1988 (March). A New CAD Method and Associated Architecture for Linear Controllers. *IEEE Trans. Automatic Control*. Vol. 33, No. 3, 1988.
- [Colgate and Hogan 87a] J. E. Colgate and N. Hogan, 1987. On the Stability of a Manipulator Interacting with its Environment. *Proc. 25th Annual Allerton Conf. Communication, Control and Computing*.
- [Colgate and Hogan 87b] J. E. Colgate and N. Hogan, 1987 (July). Robust Control of Dynamic Interacting Systems. *Int. J. Control*.
- [Elosegui, Daniel and Sharkey 90] P. Elosegui, R. Daniel and P. Sharkey, 1990. Joint Servoing for Robust Manipulator Force Control. *Proc. 1990 IEEE Int. Conf. Robotics and Automation*.
- [Eppinger and Seering 86] S. D. Eppinger and W. P. Seering, 1986. On Dynamic Models of Robot Force Control. *Proc. 1986 IEEE Int. Conf. Robotics and Automation*.
- [Eppinger and Seering 87] S. D. Eppinger and W. P. Seering, 1987. Understanding Bandwidth Limitation in Robot Force Control. *Proc. 1987 IEEE Int. Conf. Robotics and Automation*.
- [Eppinger and Seering 88] S. D. Eppinger and W. P. Seering, 1988. Modelling Robot Flexibility for End-Point Force Control. *Proc. 1988 IEEE Int. Conf. Robotics and Automation*.
- [Eppinger and Seering 89] S. D. Eppinger and W. P. Seering, 1989. Three Dynamic Problems in Robot Force Control. *Proc. 1989 IEEE Int. Conf. Robotics and Automation*.
- [Fahmy and O'Rerlly 82] M. M. Fahmy and J. O'Rerlly, 1982 (June). On Eigenstructure Assignment in Linear Multivariable Systems. *IEEE Trans. Automatic Control*. Vol. AC-27, No.3, pp. 690-693.

- [Fukuda, Kitamura and Tanie 87] T. Fukuda, N. Kitamura and K. Tanie, 1988. Adaptive Force Control of Manipulator with Consideration of Object Dynamics. *Proc. 1987 IEEE Int. Conf. Robotics and Automation*.
- [Goldenberg 88] A. A. Goldenberg, 1988. Implementation of Force and Impedance Control in Robot Manipulators. *Proc. 1988 IEEE Int. Conf. Robotics and Automation*. pp. 1626-1632.
- [Hamilton 88] W. E. Hamilton Jr., 1988. Globally Stable Compliant Motion Control for Robotics Assembly. *Proc. 1988 IEEE Int. Conf. Robotics and Automation*.
- [Hemami and Wyman 79] H. Hemami and B. Wyman, 1979. Indirect Control of the Force of Constraint in Dynamic Systems. *ASME J. Dynamic Systems, Measurement, and Control*. Vol. 101, PP. 355-360.
- [Hogan 85] N. Hogan, 1985 (March). Impedance Control: An Approach to Manipulation. Part 1. Theory, Part 2. Implementation, Part 3. Application. *ASME J. Dynamic Systems, Measurement, and Control*. Vol. 107, pp. 1-24.
- [Hogan 87] Hogan, 1987. Beyond Regulators: Modelling Control Systems as Physical Systems. *1987 American Control Conf.* pp. 1468-1476.
- [Honehangi and Koivo 87] N. Honehangi and A. J. Koivo, 1987. Eigenvalue Assignment and Performance Index Based Force-Position Control with Self-Tuning for Robotic Manipulators. *Proc. 1987 IEEE Int. Conf. Robotics and Automation*. pp. 1386-1391.
- [Huang 86] H. P. Huang, 1986. Constrained Manipulators and Contact Force Control of Contour Following Problems. *Ph.D. thesis, University of Michigan, Dept. of Electrical Engineering and Computer Science*.
- [Huang and McClamroch 87] H. P. Huang and N. H. McClamroch, 1987. Time-Optimal Control of Robotic Contour Following Problem. *Proc. 1987 IEEE Int. Conf. Robotics and Automation*.

- [Ishikawa, Sawadu, Kawase and Takata 89] H. Ishikawa, C. Sawadu, K. Kawase and M. Takata, 1989. Stable Compliance Control and Its Implementation for a 6 D. O. F. Manipulator. *Proc. 1989 IEEE Int. Conf. Robotics and Automation*.
- [Jury 56] E.I. Jury, 1956 (November). Synthesis and Critical Study of Sampled-Data Systems. *AIEE Trans.*, 71, pt. 2, 141-51.
- [Khatib and Burdick 86] O. Khatib and J. Burdick, 1986. Motion and Force Control of Robot Manipulators. *Proc. 1986 IEEE Int. Conf. Robotics and Automation*. pp. 1381-386.
- [Kazerooni, Sheridan and Honpt 86] H. Kazerooni, T. B. Sheridan and P. K. Honpt, 1986. Robust Compliant Motion for Manipulators. Part 1. The Fundamental Concept of Compliant Motion, Part 2. Design Method. *IEEE J. Robotics and Automation*. Vol. 1, No. 2, pp. 83-105.
- [Kazerooni 87] H. Kazerooni, 1987. Robust, Non-Linear Impedance Control for Robot Manipulators. *Proc. 1987 IEEE Int. Conf. Robotics and Automation*. pp. 741-750.
- [Kazerooni and Tsay 1988a] H. Kazerooni and T. I. Tsai, 1988. Compliance Control and Unstructured Modelling of Cooperating Robots. *Proc. 1988 IEEE Int. Conf. Robotics and Automation*. pp. 510-151.
- [Kazerooni and Tsay 1988b] H. Kazerooni and T. I. Tsai, 1988. Stability Criteria for Robot Compliance Maneuvers. *Proc. 1988 IEEE Int. Conf. Robotics and Automation*. pp. 1166-1172.
- [Koivo 86] A. J. Koivo, 1986. Force-Position-Velocity Control with Self-Tuning for Robotic Manipulators. *Proc. 1986 IEEE Int. Conf. Robotics and Automation*. pp. 1563-1568.
- [Lawrence 88] D. A. Lawrence, 1988. Impedance Control Stability Properties in Common Implementations. *Proc. 1988 IEEE Int. Conf. Robotics and Automation*.

- [Liu, Chang and Zhang 88] M-H. Liu, W-S. Chang and L-Q. Zhang, 1988. Dynamic and Adaptive Force Controller for Robotic Manipulators. *Proc. 1988 IEEE Int. Conf. Robotics and Automation*. pp. 1478-1483.
- [Mason 81] M. Mason, 1981. Compliance and Force Control for computer Controlled Manipulators. *IEEE Trans. Systems, Man and Cybernetics*. SMC-11 (6): 418-432
- [McClamroch and Huang 85] N. H. McClamroch and H. P. Huang, 1985 (June). Dynamics of Closed-Chain Manipulator. *1985 American Control Conf.*
- [McClamroch 86] N. H. McClamroch, 1986. Singular Systems of Differential Equations as Dynamic Models for Constrained Robot Systems. *Proc. 1986 IEEE Int. Conf. Robotics and Automation*.
- [Moore 81] B.C. Moore, 1981 (February). Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction. *IEEE Trans. Automatic Control*. Vol. AC-26, No.1
- [Nevins and Whitney 73] J. L. Nevin and D. E. Whitney, 1973 (September). The Force Vector Assembler Concept. *Proc. 1st CISMIFTOMM Symposium on Theory and Practice of Robots and Manipulators*.
- [Orin and Oh] D. E. Orin and S. Y. Oh, 1981 (June). Control of Force Distribution in Robotic Mechanisms Containing Closed Kinematic Chains. *ASME J. Dynamic Systems, Measurement, and Control*. Vol. 103, pp. 134-141.
- [Paul and Shimano 76] R. P. Paul and B. Shimano, 1976 (July). Compliance and Control. *Proc. Joint Automatic Control Conf.* pp. 694-699.
- [Raibert and Craig 81] M.H. Raibert and J. J. Craig, 1981 (June). Hybrid Position/Force Control of Manipulators. *ASME J. Dynamic Systems, Measurement, and Control*. Vol. 103, pp.126-133.
- [Salisbury 80] J. K. Salisbury, 1980. Active Stiffness Control of a Manipulator

in Cartesian Coordinates. *Proc. 1980 IEEE Conf. Decision and Control*. pp. 95-100.

[Schutter 88] J. De Schutter, 1988. Improved Force Control Laws for Advanced Tracking Application. *Proc. 1988 IEEE Int. Conf. Robotics and Automation*. pp. 1497-1502.

[Sharon, Hogan and Hardt 88] A. Sharon, N. Hogan and D. E. Hardt, 1988. High Bandwidth Force Regulation and Inertia Reduction Using a Macro/Micro Manipulator System. *Proc. 1988 IEEE Int. Conf. Robotics and Automation*.

[Shin and Lee 85] K. G. Shin and C-P. Lee, 1985. Compliant Control of Robot Manipulators with Resolved Acceleration. *Proc. 24th IEEE Conf. Decision and Control*. Ft. Lauderdale. pp. 350 -357.

[Slotine 85] J. J. Slotine, 1985. The Robust Control of Robot Manipulators. *Int. J. Control*. 4, 2.

[Slotine and Li 87a] J. J. Slotine and W. Li, 1987. Adaptive Manipulator Control: Parameter Convergence and Task-Space Strategies. *Proc. 1987 IEEE Int. Conf. Robotics and Automation*. pp. 828-835.

[Slotine and Li 87b] J. J. Slotine and W. Li, 1987. Adaptive Strategies in Constrained Manipulators. *Proc. 1987 IEEE Int. Conf. Robotics and Automation*. pp. 595-601.

[Slotine and Sastry 83] J. J. Slotine and S. S. Sastry, 1983. Tracking Control of Non-Linear Systems Using Sliding Surfaces, with Application to Robot Manipulators. *Int. J. Control*. 38, 465

[Tarn, Bejczy and Yun 88] T. J. Tarn, A. K. Bejczy and X-P. Yun, 1988. Robot Arm Force Control Through System Linearization by Nonlinear Feedback. *Proc. 1988 IEEE Int. Conf. Robotics and Automation*.

[Verl 89] M. Verl, 1989. Development of a Torque Controller for the Adep-

tOne Robot. *M.S. thesis, Case Western Reserve University, Dept. of Electrical Engineer and Applied Physics.*

[Vidyasagar 85] M. Vidyasagar, 1985. Control System Synthesis: A Factorization Approach. *The MIT Press, 1985.*

[Wang and Kuo 88] S-T. Wang and C. Y. Kuo, 1988. Nonlinear Robust Hybrid Control of Robot Manipulators. *Proc. 1988 IEEE Int. Conf. Robotics and Automation. pp. 1204-1209.*

[West and Asada 85] H. West and H. Asada, 1985 (June). A Method for the Control of Robot Arm Constrained by Contact with the Environment. *Proc. 1985 American Control Conf.*

[Whitney 77] D. E. Whitney, 1977 (June). Force Feedback of Manipulator Fine Motions. *ASME J. Dynamic Systems, Measurement, and Control.*

[Whitney 87] D. E. Whitney, 1987. Historical Perspective and State of the Art in Robot Force Control. *Int. J. Robotics Research. 6(1): 3-14.*

[Wlassich 86] J. J. Wlassich, 1986. Nonlinear Force Feedback Impedance Control. *M.S. thesis, MIT, Dept. of Mechanical Engineering.*

[Wu and Paul 80] C. H. Wu and R. Paul, 1980. Manipulator Compliance Based on Joint Torque Control. *1980 IEEE Conf. Decision and Control.*

[Xu and Paul 88] Y. Xu and R. P. Paul, 1988. On Position Compensation and Force Control Stability of a Robot with a Compliant Wrist. *Proc. 1988 IEEE Int. Conf. Robotics and Automation. pp. 1173-1178*

[Yabuta, Chonu and Beni 88] T. Yabuta, A. J. Chonu and G. Beni, 1988. *Proc. 1988 IEEE Int. Conf. Robotics and Automation.*

[Yoshikawa 86] T. Yoshikawa, 1986. Dynamic Hybrid Position/Force Control of Robot Manipulators—Description of Hand Constraints and Calculation of

Joint Driving Force. *Proc. 1986 IEEE Int. Conf. Robotics and Automation*. pp. 1393-1398.

[Youcef-Toumi and Li 87] K. Youcef-Toumi and D. Li, 1986. *Proc. 1987 IEEE Int. Conf. Robotics and Automation*. pp. 2055-2060.

[Zames 66] G. Zames, 1966 (April). On the Input-Output Stability on Non-linear Time-Varying Feedback Systems – Part 1: Conditions Derived Using Concepts of Loop Gain, Conicity, and Positivity. *IEEE Trans. Automatic Control*, AC - 11:228-238, 1966.

[Zhang 89] H. Zhang, 1989. Kinematic Stability of Robot Manipulators under Force Control. *Proc. 1989 IEEE Int. Conf. Robotics and Automation*.